DLR-IB-RM-OP-2017-140

Pose Estimation and Traction Control for a Planetary Exploration Rover

Masterarbeit

Lukas Meyer

Deutsches Zentrum
DLR für Luft- und Raumfahrt

Institut für Robotik und Mechatronik	BJ.: 2017
	IB.Nr.: DLR-IB-RM-OP-2017-140

MASTER'S THESIS

POSE ESTIMATION AND TRACTION CONTROL FOR A PLANETARY EXPLORATION ROVER

Freigabe:	Der Bearbeiter:	Unterschriften
	Lukas Meyer	
	Betreuer:	
	Kristin Bussmann	
	Der Institutsdirektor	
	Prof. Dr. Alin Albu-Schäffer	

Dieser Bericht enthält 86 Seiten, 27 Abbildungen und 2 Tabellen

Ort: Oberpfaffenhofen	Datum:	Bearbeiter:	Zeichen:

Lehrstuhl für Raumfahrttechnik
Prof. Prof. h.c. Dr. Dr. h.c.
Ulrich Walter

Technische Universität München

Master's Thesis

Pose Estimation and Traction Control for a Planetary Exploration Rover

LRT-MA 2017/08

Author:

Lukas Meyer

Supervisors: Kristin Bussmann, M.Sc.
Dr.-Ing. Alexander Dietrich
Institut für Robotik und Mechatronik
Deutsches Zentrum für Luft- und Raumfahrt

Dipl.-Ing. Christian Bühler
Prof. Prof. h.c. Dr. Dr. h.c. Ulrich Walter
Lehrstuhl für Raumfahrttechnik / Institute of Aeronautics
Technische Universität München

Zusammenfassung

Planetare Rover sind ein Schlüsselelement bei der Erforschung von Planetenoberflächen in unserem Sonnensystem, wobei immer häufiger autonome, robotische Systeme zum Einsatz kommen. Die Lightweight Rover Unit (LRU) ist ein Prototyp für solch einen autonomen planetaren Rover und wurde am Institut für Robotik und Mechatronik des Deutschen Zentrums für Luft- und Raumfahrt entwickelt. Die LRU wurde speziell für die Anwendung in unwegsamen Gelände konzipiert und ist in der Lage, sowohl die Erkundung der Planetenoberfläche als auch die Manipulation von Objekten autonom durchzuführen.

Um autonom navigieren zu können, muss der Rover in der Lage sein, seine Pose im Gelände genau zu bestimmen. Dafür werden die Informationen vieler verschiedener Sensoren berücksichtigt und mithilfe eines erweiterten Kalman Filters (EKF) fusioniert. Dabei werden unter anderem die Informationen der Räder verwendet, die sogenannte Radodometrie. Die Aufgabe dieser Masterarbeit ist die Entwicklung einer dreidimensionalen Radodometrie die zudem den Radschlupf mitberücksichtigt und somit eine Verbesserung der Schätzung der Pose ermöglicht.

Die Grundlage für die Radodometrie ist ein kinematisches Modell des Rovers, welches in dieser Arbeit zuerst hergeleitet und anschließend kalibriert wird. In der Kalibrierung werden die tatsächlichen geometrischen Abmessungen des Rovers bestimmt, Sensorsignale ausgewählt und die Elastizität von Systemkomponenten modelliert.

Vor allem bei Fahrten in unwegsamem Gelände kommt es zu Schlupf, der für eine korrekte Positionsbestimmung mitberücksichtigt werden muss. Hierfür wird eine Schlupfschätzung verwendet, die den Schlupf als Fehler in der Geschwindigkeitsmessung modelliert und das Modell untergrundabhängig parametrisiert. Der bestimmte Schlupf wird zusätzlich für eine Traktionsregelung verwendet, welche zu einem verbesserten Fahrverhalten führt. Hierfür werden die Geschwindigkeit und die Beschleunigung des Rovers in Abhängigkeit des geschätzten Schlupfs angepasst.

Die LRU war Bestandteil einer Mond-Analog-Mission auf dem Vulkan Ätna in Italien. Dabei wurde das Szenario eines Robotereinsatzes auf der Mondoberfläche nachgestellt, wobei der Rover unter anderem in unwegsamem Gelände operierte. Die Konzepte und Methoden dieser Masterarbeit wurden im Hinblick auf dieses Testszenario entwickelt und auf dem Vulkan in Experimenten erfolgreich validiert. Speziell das Schlupfmodell wurde für den vulkanischen Untergrund parametrisiert und das Ergebnis anhand eines Fahrversuchs von etwa einem Kilometer Länge bestätigt.

Abstract

Planetary rovers are one of the key elements for extraterrestrial on-surface exploration in our solar system, increasingly relying on autonomous robots. The Lightweight Rover Unit (LRU) developed by the Robotics and Mechatronics Center (RMC) of the German Aerospace Center (DLR) is a terrestrial prototype of an autonomous planetary exploration rover. It features autonomy both in exploration and manipulation of the environment and is designed specifically for rough terrain locomotion.

Successful autonomous navigation requires correct knowledge of the current rover pose for which the LRU fuses the information of several sets of sensors using an extended Kalman filter. This includes the information obtained from the wheels: the wheel odometry. This master's thesis focuses on implementing a slip-aware three-dimensional wheel odometry that allows to improve the overall pose estimate.

The computation of the wheel odometry requires a kinematic model of the rover, which is derived and calibrated. The calibration process includes correction of the geometrical measures, selection of sensor signals, and modeling of the elasticity of system components.

In rough terrain locomotion, the major error source for wheel odometry is wheel slip, degrading the accuracy of the pose estimate. The thesis provides a method to account for wheel slip and shows its successful application in experiments. At first, methods for direct slip measurement are briefly investigated. As these turn out to be not applicable, a method for slip estimation utilizing a velocity-error-model is used instead. The velocity-error-model is terrain-dependent, a dependency which is modeled by parameters and is determined experimentally.

The obtained information on slip is additionally used to improve the driving performance of the LRU. Therefore, a concept for traction control is presented that reduces acceleration and velocity of the rover according to the detected slip. This concept is validated by showing the improved driving performance in experiments.

The LRU was part of a test campaign on the volcano Mt. Etna in Italy that emulated lunar on-surface operation. There, the volcanic rough terrain environment allowed the rover to fully demonstrate its off-road capabilities. The master's thesis was written in the context of this moon analogue mission and most experiments were performed on-site. The slip model is parameterized for the volcanic soil and is validated during a long-range driving test on Mt. Etna, featuring a trajectory close to one kilometer in length.

Contents

List of Figures

List of Symbols

System Modeling

$\boldsymbol{C}$	covariance matrix
$\boldsymbol{F}$	dynamic matrix in $\dot{\boldsymbol{x}} = \boldsymbol{F}\boldsymbol{x} + \boldsymbol{G}\boldsymbol{u}$
$\boldsymbol{G}$	input coupling matrix
$\boldsymbol{H}$	measurement matrix in $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x}$
$\boldsymbol{P}$	error-covariance matrix of $\tilde{\boldsymbol{x}}$
$\boldsymbol{Q}$	covariance matrix of the process noise $\boldsymbol{w}$
$\boldsymbol{R}$	covariance matrix of the measurement noise $\boldsymbol{v}$
$\boldsymbol{\Phi}$	state transition matrix
$\boldsymbol{v}$	measurement noise
$\boldsymbol{w}$	noise in the dynamic system
$\boldsymbol{x}$	state vector
$\hat{\boldsymbol{x}}$	state estimate
$\tilde{\boldsymbol{x}}$	estimation error $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \hat{\boldsymbol{x}}$

Geometry and Kinematics

$\boldsymbol{H}_{\mathrm{AB}}$		homogeneous transformation between frames A and B
$\boldsymbol{R}_{\mathrm{AB}}$		rotation between frames A and B
$\boldsymbol{T}_{\mathrm{AB}}$		combined transformation of linear and angular velocities
$\boldsymbol{W}_{\mathrm{AB}}$		transformation of angular velocities
$\boldsymbol{p}_{\mathrm{AB}}$	$[\mathrm{m}]$	translation between frames A and B
$\boldsymbol{a}_{\mathrm{B}}$	$[\mathrm{m/s^2}]$	linear acceleration of a body B
$\boldsymbol{v}_{\mathrm{B}}$	$[\mathrm{m/s}]$	linear velocity of a body B
$\boldsymbol{\omega}_{\mathrm{B}}$	$[^\circ/\mathrm{s}]$	angular velocity of a body B

$\boldsymbol{\zeta}$		6 DOF pose
$\boldsymbol{\xi}$		3 DOF pose
α	$[^\circ]$	roll angle
β	$[^\circ]$	pitch angle
γ	$[^\circ]$	yaw angle
θ	$[^\circ]$	yaw angle for the 3 DOF pose
l_x	$[\mathrm{m}]$	body length, body center to wheels
l_y	$[\mathrm{m}]$	body width, body center to wheels
l_z	$[\mathrm{m}]$	vertical distance from body center to wheels
r	$[\mathrm{m}]$	wheel radius

General Symbols

$\boldsymbol{H}_{\mathrm{sys}}$		slip parameter Jacobian matrix
$\boldsymbol{I}$		identity matrix
$\boldsymbol{g}$	$[\mathrm{m/s^2}]$	gravity vector
$\boldsymbol{p}$		slip parameter vector
$E(*)$		mean value of $*$
F_t	$[\mathrm{N}]$	tractive force at a wheel
d	$[\mathrm{m}]$	distance
i		wheel index
k		time step, update step
m	$[\mathrm{kg}]$	mass
R	$[\mathrm{N}]$	resistive force at a wheel
s		slip
T	$[\mathrm{Nm}]$	wheel torque
ν		update rate
φ_i	$[^\circ]$	steering angle of wheel i
ρ	$[^\circ]$	bogie angle
ψ_i	$[^\circ]$	wheel contact angle
ω_i	$[^\circ/\mathrm{s}]$	angular velocity of wheel i

List of Abbreviations

DGPS	differential global positioning system
DLR	German Aerospace Center
DOF	degrees of freedom
EKF	extended Kalman filter
IMU	inertial measurement unit
IPEM	Integrated Prediction Error Minimization
LRT	long range test
LRU	Lightweight Rover Unit
MER	Mars Exploration Rover
MSL	Mars Science Laboratory
RMC	Robotics and Mechatronics Center
ROBEX	Robotic Exploration of Extreme Environments
ROS	Robot Operating System
SEA	serial elastic actuator
SLAM	simultaneous localization and mapping
WMR	wheeled mobile robot

1. Introduction

The understanding of our solar system and space in general is a task which fascinated humanity since ages. However, over centuries and millennia, the analysis of space was limited to astronomical observations only. Due to the technological advances during the last century, not only observation but also exploration of space was made possible, suddenly providing new means of research. The new technologies allowed to send probes and even humans into space, widening the understanding of our solar system.

A multitude of robotic spacecrafts was sent into space designated for many different objectives, with the main focus of exploration oriented towards the nearest neighbors of the Earth. Especially considering the three astronomical bodies closest to Earth: the Moon and the planets Mars and Venus.

Particularly Moon and Mars where subject to many on-surface missions with the Apollo program being the most prominent one – the only mission where humans ever set foot on extraterrestrial soil [1, 2].

All other exploration missions on the surfaces of Moon and Mars were performed by landing stationary probes or even mobile vehicles – usually referred to as lander and rover respectively – mostly teleoperated from earth. Non-human robotic missions are advantageous, as they allow to collect data on the surfaces of Moon and Mars in circumstances which are deemed to be too complicated, too dangerous, or too expensive to be done by humans.

Teleoperation of mobile vehicles over the distance of several thousands or millions of kilometers include many challenges like communication delay and bandwidth limitation, which make the use of partial or even fully autonomous systems necessary [3].

The Lightweight Rover Unit (LRU) developed by the Robotics and Mechatronics Center (RMC) of the German Aerospace Center (DLR) is a terrestrial prototype of an autonomous planetary exploration rover aiming to overcome mentioned challenges [4]. It was developed with the focus on planetary exploration missions, mainly for the moon, but could also be used in terrestrial search and rescue scenarios [4]. The LRU takes part in the Robotic Exploration of Extreme Environments (ROBEX) project alliance, in which context a moon analogue demonstration mission was performed on Mt. Etna in Sicily, Italy during June and July 2017 [5]. This master's thesis was written in the context of the moon analogue mission and most experiments were performed on Mt. Etna.

1.1. Motivation

The key aspect of the LRU is its autonomy in both exploration and manipulation for tasks in rough terrain under alien conditions, while additionally possessing a lightweight structure for payload reduction [6]. For the autonomous navigation, correct information on the position of the rover is crucial. Extraterrestrial environments lack accurate global positioning systems and therefore need to rely on other methods for self-localization [7].

In the case of the LRU, self-localization is especially important, as the rover relies on simultaneous localization and mapping (SLAM) for navigation [6]. In unknown terrain, no precise information on the surface and on potential obstacles exists. SLAM is a technique which allows for autonomous map building during robotic exploration of the terrain. There, the rover uses its sensoric system for feature and obstacle detection relative to its position, which are incorporated into a local map [8, p. 250]. Subsequently, this local map allows for navigation by recognizing previously detected features. However, a very precise self-localization is required as the created map and the estimated position correlate: an imprecise position estimate will result in an imprecise localization of the detected features for the map, which in return degrade the accuracy of the next position estimate [8, p. 250].

Usually, planetary rover posses a multitude of different sensors that can be used for self-localization, where each type of sensor is expected to have a certain error in its information. To determine the *pose* of the rover – orientation (attitude) and position combined [9, p. 9] – the information of multiple sensors is merged together, trying to cancel out the individual sensor errors. This process is called *sensor fusion* and tries to provide a pose estimate, where the resulting error is smaller than the error of the best input provided by an individual sensor [8, p. 227].

For localization, all wheeled mobile robots (WMRs) have one possible set of sensors always available: the wheels. Measuring the angular velocities of the wheels in rolling direction together with the steering angles allows to determine the body velocity of the rover. This form of pose estimation is also referred to as *wheel odometry*.

Using wheel odometry includes several challenges. To calculate the body velocity, a kinematic model of the rover is necessary that allows to relate the velocities of the wheels to the body. An imprecise knowledge of the system geometry, its sensor accuracy and the dynamic system behavior result in erroneous information on the position and has to be corrected by adequate calibration. The locomotion in rough terrain provides another challenge for the wheel odometry: Wheels tend to slip in loose soil, therefore overestimating the distance driven by the rover. To provide a correct wheel odometry in rough terrain, measurement or estimation of the occurring slip is essential.

The slip of wheels does not only need to be considered for pose estimation. Instead, if information on slip is already available, it can be used for traction control to improve the rough terrain locomotion of the rover.

This thesis addresses the mentioned topics and introduces the wheel odometry, slip estimation, and traction control for the LRU.

1.2. State of the Art

There exist a multitude of planetary exploration rovers, each having different sets of sensors, numbers of wheels, and an unique kinematic structure. This variety requires different methods for pose estimation and traction control for each system and limits the comparability between the systems.

Some of the most prominent rovers are the Mars rovers from the Mars Exploration Rover (MER) and Mars Science Laboratory (MSL) family, each having one rover currently operating on Mars – Opportunity and Curiosity respectively [2]. Each of the rovers is able to perform traversing maneuvers on the surface of Mars autonomously using camera information for hazard detection and path planning [3, 7]. The MER and MSL rovers both use similar approaches to determine their pose. Mainly, the inertial measurement unit (IMU) and wheel odometry information are used, additionally supported by visual odometry in the case of wheel slippage [7]. Furthermore, measurements of the sun position relative to the rover is occasionally used to correct errors in the attitude estimate [7].

To this point, no other active rovers operate outside of earth. However, a multitude of other rover systems exist. Some were used during previous missions, like the Sovjet and Chinese lunar rovers Lunokhod [10] and Yutu [11] respectively and some as prototypes for future missions like the European ExoMars Rover [12] or the American Mars 2020 rover [2]. Additionally, many terrestrial platforms exist for experimental purposes, for example the rovers Rocky 7 [13], Rocky 8 [14], Zoë [15], Shrimp [16], Micro6 [17], or Fluffy [18].

A detailed framework for the two-dimensional kinematic modeling of rovers is introduced by [19], which is augmented to full three dimensions by [13], [20], or [21]. Work on vehicle localization by sensor fusion can be found amongst others in [22], [23], or [17]. There, the sensor fusion relies on different enhanced Kalman filtering techniques: the extended Kalman filter, the information filter, and the unscented Kalman filter respectively. A concept for sensor fusion and slip compensation, which is similar to the one used on the LRU, is described in [14]. Information on odometry calibration is provided for example in [24] or [25]. A wheel odometry error model can be found in [26] and [15].

A lot of work has been done in the area of slip detection and slip estimation for planetary rovers. This area is especially dependent on the rover structure, its sensors, and its mode of operation. All the system specific factors define, if

slip can be measured directly or needs to be estimated instead. Work on slip detection and estimation has been done by [14], [27], and [18] amongst others.

If knowledge on the occurring slip exist, it can additionally be used for traction control, for example in [28], [29], [16] or [30].

Slip detection and traction control is also considered in several other fields, especially for agricultural and forestry machines (See for example [31]). Note that methods for slip from automotive research cannot be applied here, as these usually consider slip as result of highly dynamic maneuvers on paved roads.

Other work on traction control can be found for example in [32], which focuses less on slip but rather on wheel contact force maximization.

1.3. Definition of Tasks

The main task of this master thesis is to improve the overall pose estimation of the LRU by increasing the accuracy of the wheel odometry. To achieve this objective, many different aspects need to be considered and the rover system needs to be described.

A detailed description of the LRU and the system components is presented in chapter 2, where both the hardware and software are considered. The LRU is a complex system and the role of the wheel odometry in the system architecture and the interaction with the other system components is additionally outlined. Furthermore, the general concept for pose estimation is described.

Chapter 3 provides a short overview on the mathematical methods used for modeling the system and estimating the pose.

To improve the information provided by the wheel odometry, several tasks are necessary. In chapter 4, a three-dimensional kinematic model is presented which allows to determine the body velocity by measuring wheel velocities, the steering angles, and joint angles. The same chapter briefly discusses additional topics on wheel odometry: the noise in sensor signals and simplifying modeling assumptions. Additionally, the first major error source in wheel odometry is addressed: overcoming an incorrect calibration. The applied calibration strategies and the derived wheel odometry from the kinematic model are validated in laboratory driving experiments

Chapter 5 considers the other main influence on the error in wheel odometry: the slipping of the wheels. The theoretical background for slip and wheel-soil interaction are briefly introduced at the beginning of the chapter. At first, some potential approaches for direct slip measurement are introduced and explained why these are not feasible on the LRU. As the system provides no means for direct slip measurement, a method for slip estimation is presented instead, which allows to estimate the slip using a parameterized, terrain-dependent velocity-error-model. The parameters for the slip-model are determined experimentally

for the Etna soil and are subsequently validated by a long-range driving test during the 2017 ROBEX Moon Analogue Mission.

The final part of the thesis, chapter 6, considers traction control. The information on the slip of the rover together with the theory of wheel-ground-interaction in loose soil is used to determine feed-forward control laws which allow for an improved driving performance of the rover in rough terrain.

1.4. ROBEX Moon Analogue Mission

This thesis was written in the context of the ROBEX Moon Analogue Mission on Mt. Etna. The mission aims to demonstrate the key aspects of a possible lunar mission scenario, where a seismic sensor network is installed on the surface of the Moon, similar to the Apollo 17 mission [33]. A lunar lander transports the seismic sensor boxes as payload to the surface of the moon. These are to be extracted by the mobile autonomous robot using robotic manipulation and then distributed over the surface of the moon [33]. The rover needs to fulfill two key tasks during this mission: first, it has to show mobility and autonomous navigation during the drives in rough terrain. Second, it needs to demonstrate successful manipulation and precise positioning of the seismic sensors during extraction from the lander and while deploying them on the lunar surface [33].

The selected test site for the demonstration mission is at the southern flank of Mt. Etna at an altitude of approximately $2600\,\mathrm{m}$. It was selected due to its similarity with the lunar surface and due to the constant seismic activity of the volcano, which allows for frequent sensoric input to the seismic boxes [34]. The main mission took place during June and July 2017 with a previous preparatory field test in September 2016 [34].

The first tests and measurements for this master thesis were performed at the RMC institute, but most of the experiments took place on Mt. Etna during the demo mission.

2. System Overview

In this chapter, a brief overview on the LRU is given. On one hand, the focus lays on the mechanical components, the software architecture and the different possible configurations. On the other hand, the system is described with regard to pose estimation.

2.1. System Components and Locomotion Concept

The LRU is currently available in two different configurations. The components identical for both configurations are outlined within this section and the configuration specific aspects are later listed in section 2.2. One of the rovers – the LRU-2 – is shown in figure 2.1. The length of both rovers is $1080\,\mathrm{mm}$ and the width $730\,\mathrm{mm}$ and allows for a maximum driving velocity of $1.1\,\mathrm{m/s}$ in rough terrain [6]. The main body contains several on-board computers, the IMU, and the batteries amongst other components. The pan-tilt camera system for visual odometry is mounted on top of the body [6].

fig. 2.1.: The LRU-2 equipped with the robotic arm and holding up one seismic sensor-box during the 2016 field test on Mt. Etna. © DLR

The bogie frames are the structure connecting the wheels to the body. They are mounted at the front and the rear of the body via a serial elastic actuator (SEA). The SEA is a motor-spring-damper system which allows for passive suspension and active rotation along the longitudinal body axis [4]. Two wheels are attached to the front and rear bogies each and all are equipped with a steering and a driving motor, a setup that allows to drive and steer all of the four wheels individually [6]. The driving motors are mounted directly in the wheel hubs and the steering motors are mounted at each end of the bogie frames, directly above the wheels.

This design of the wheels allows for two different drive modes of the rover. In the first mode, the wheels of the rover align in such way that the axis of the wheel hub axles intersect in a common point – thus providing the same instantaneous center of rotation for each wheel. The rover is therefore able to move with a forward velocity and an angular velocity in its current plane of motion. For the other driving mode, all four wheels align in parallel to each other, thus allowing for a linear movement of the rover in any direction in its current plane of motion while maintaining a constant orientation. During this work, both driving modes are referred to as *ackermann* or *crab mode* respectively.

The control of the wheels is achieved by the so called *kinematic control*. The LRU control receives a command for the desired velocity of the rover body, either from the autonomous software components or via a joystick. The inverse kinematics algorithm calculates the wheel speeds and steering angles which are necessary to achieve the desired body velocity. The resulting positions and velocities are commanded to the motor controllers for steering a driving respectively. Note, that the two mentioned driving modes result from mapping the desired velocities to the wheels, where either longitudinal together with lateral linear velocities or longitudinal together with angular velocities are considered. This distinction is merely a design choice and could be omitted in the future, considering all three velocity components simultaneously.

To accomplish the complex tasks assigned to the LRU, a suited software architecture is required. Three different middlewares run on the LRU: the open source *Robot Operating System (ROS)* and the two DLR internal *Links and Nodes* and *SensorNet* [6]. SensorNet processes and distributes the vision data, Links and Nodes is a process manager, amongst others responsible for the real-time control, and ROS is used to connect the higher-level software components [6].

The wheel odometry in this thesis is implemented within the real-time controller and communicates with the hardware via the in-house-developed software *robotkernel*. The controller model is designed in Simulink® and compiled to C-code, with its execution managed by Links and Nodes.

The sensor fusion for pose estimation is performed by an extended Kalman filter (EKF), which collects the sensor data from the IMU, the wheel odometry, and the visual odometry via ROS [6]. The information computed by the wheel

odometry is provided by Links and Nodes to ROS and gets published there as so-called *topics*.

2.2. Rover Configurations

The LRU is a mobile robotic platform suited for many different applications. During the 2017 Etna mission (see section 1.4), two LRU units with different configurations were in operation.

Both units are equipped with identical body, bogie, and wheel components, except for different spokes on the wheels.

The LRU-1 is equipped with a camera system referred to as *science cam* based on the concept proposed in [33]. The science cam offers different cameras for geological observations additionally to the stereo-camera system for navigation and is depicted on the LRU-1 in figure 5.4. The LRU-2 is equipped with a much smaller pan-tilt stereo camera, solely for navigation, together with a robotic arm for manipulation [6] and can be seen in figure 2.1.

The different configurations have little influence on the wheel odometry, except for the different spoke configurations of the wheels and a different mass and mass-distribution.

During this thesis, the experiments had to be performed on different systems due to limitations in system time. For each experiment, the used system is denoted and it is assumed that the results can be transferred to the other system applying minor adaptations only.

2.3. Architecture for Pose Estimation

The pose estimation is a crucial part of the robot navigation. The concept for pose estimation is outlined in figure 2.2, which additionally presents the elements for path planning, inverse kinematics and traction control.

The EKF used in the LRU is described in detail in [35], but fundamental information on Kalman filtering can be found in section 3.4. The architecture of the wheel odometry, slip estimation and traction control on implementation level is shown in figure 2.3.

2.3.1. Available Sensors

First of all, the available sensors are presented. Figure 2.2 shows the three major categories of sensoric input: the visual odometry (VO), the IMU, and the wheel odometry (WO).

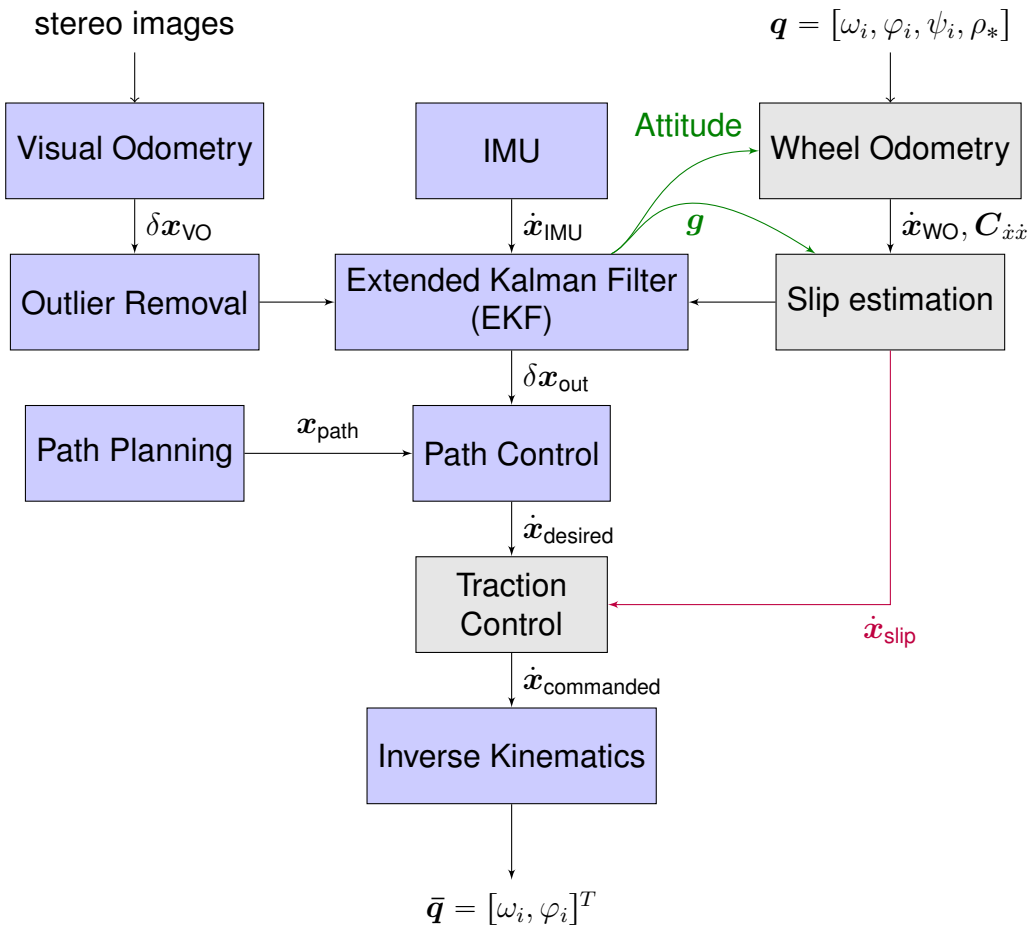

fig. 2.2.: Concept for pose estimation and traction control: the gray blocks are part of this master's thesis. Adapted from [14].

The visual odometry uses the images obtained by the stereo cameras in the pan-tilt camera system and processes them. This results in position information on objects in proximity of the LRU, which is used for SLAM. The relative position of the rover in a local reference frame is determined according to the detected features and the local map is updated. The determined position is checked for erroneous results, which are discarded by the outlier removal process, all other measurements are passed onward to the EKF [6].

The data from the IMU is incorporated into the filter as second input, providing information on the linear accelerations of the rover body and its angular velocities.

The final information source for pose estimation is the wheel odometry. It uses the kinematic model to find the forward kinematics, which allow to find the body velocity of the rover by measuring the wheel speeds [8, p. 185f]. The forward kinematics can provide velocity estimates on up to 6 degrees of freedom (DOF), depending on the used kinematic model and the observability of the individual DOF. So far, only part of the wheel odometry is integrated into the EKF fusion

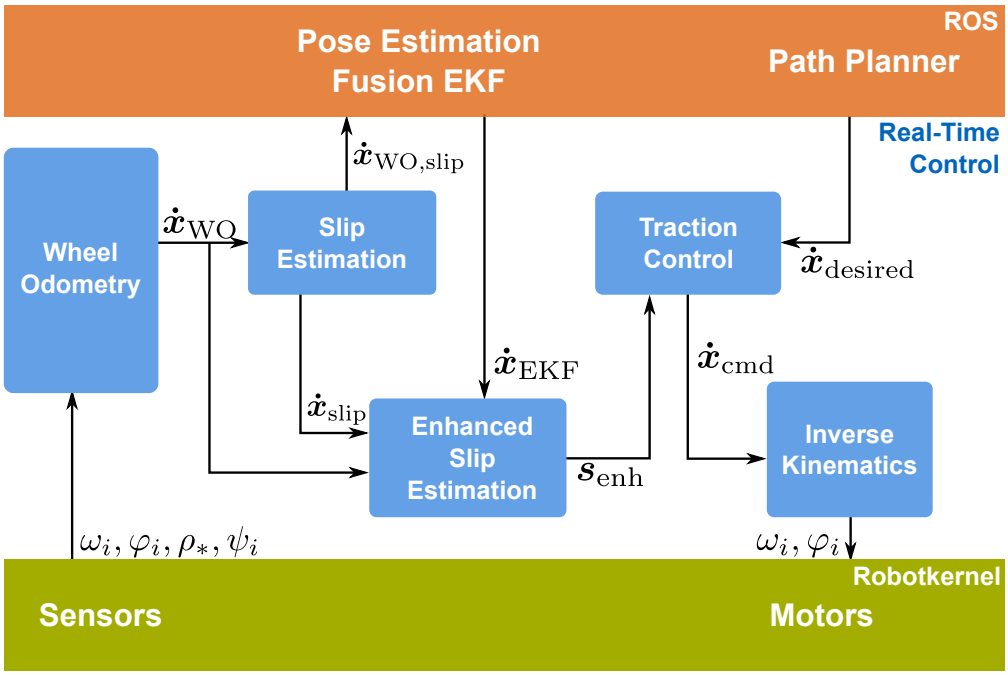

fig. 2.3.: Architecture of wheel odometry, slip estimation and traction control on implementation level.

– the linear longitudinal and lateral velocities of the rover body – but this thesis provides a framework for a possible use of further velocity components (see section 4.3.3 for more details).

All eight motors on each wheel – both for steering and driving – provide the same set of data: The encoder position, the angular velocities obtained from the encoder values, and the applied motor torque derived from the measured current. Additionally, potentiometers are available, which are mostly used to define the initial position after rover startup. The motors used in the SEAs provide the same information, augmented by the angle of the passive spring-damper suspension, measured by a potentiometer. The sensor signals are discussed in more detail in sections 4.4.2 and 4.5.

2.3.2. Wheel Odometry and Slippage

Before the wheel odometry is passed to the EKF sensor fusion, occurring slip needs to be determined. An EKF is designed to handle stochastic errors in sensor signals, but requires input signals preferably free of systematic errors (see chapter 3 for details). Wheel slip is the error between the measured and the real velocity due to insufficient contact between the wheels and the ground and is an example for a systematic error. Thus, it is essential to consider slip before passing the wheel odometry velocities to the EKF.

Figure 2.2 and 2.3 additionally outline another aspect of the thesis. The obtained information on slip is used to improve the driving performance of the

rover by adding a feed-forward traction control mechanism between the path control and the inverse kinematics, acting on the commanded body-velocities and accelerations. The traction control uses an *enhanced slip estimate*, which uses additional velocity information from the EKF output to improve the estimate. Note that the EKF information can only be used for traction control and not for the slip estimation for wheel odometry, as it would otherwise introduce strong correlations to the fusion filter.

3. Kalman Filter and Least Squares Estimation

The task of pose estimation requires the use of mathematical methods which allow to model systems, deal with uncertainties in measurements and processes, and allow to extract information from over-determined systems. This chapter introduces some of the mathematical methods used throughout this thesis and briefly discusses necessary the mathematical framework.

3.1. Dynamic Systems

Dynamic systems can be modeled mathematically by a state space description, with a notation following [36]. For a general nonlinear continuous systems, such description reads

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t), \tag{3.1a}$$
$$\boldsymbol{y}(t) = \boldsymbol{h}(\boldsymbol{x}(t), t), \tag{3.1b}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ denotes the state of the system, $\boldsymbol{u} \in \mathbb{R}^k$ is the system input and $\boldsymbol{y} \in \mathbb{R}^m$ is the output or the measurement of the system. For the case of a linear system, the equations can be written as

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(t)\boldsymbol{x}(t) + \boldsymbol{G}(t)\boldsymbol{u}(t), \tag{3.2a}$$
$$\boldsymbol{y}(t) = \boldsymbol{H}(t)\boldsymbol{x}(t), \tag{3.2b}$$

with the dynamic matrix $\boldsymbol{F} \in \mathbb{R}^{n\times n}$, the input coupling matrix $\boldsymbol{G} \in \mathbb{R}^{n\times k}$, and the measurement matrix $\boldsymbol{H} \in \mathbb{R}^{m\times n}$.

If a time-discrete system is described, the equations change to

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k\boldsymbol{x}_k + \boldsymbol{G}_k\boldsymbol{u}_k, \tag{3.3a}$$
$$\boldsymbol{y}_k = \boldsymbol{H}_k\boldsymbol{x}_k, \tag{3.3b}$$

which relates the state $\boldsymbol{x}$ at time step $k+1$ to the state and the input of the previous time step k.

Note that the system matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ are not identical for the time continuous and time discrete case but have to be transformed using the *state transition matrix* $\boldsymbol{\Phi}(t_2, t_1)$. The state transition matrix is used to express the general solution of (3.2), which reads [36]

$$\boldsymbol{x}(t_2) = \boldsymbol{\Phi}(t_2, t_1)\boldsymbol{x}(t_1) + \int_{t_1}^{t_2} \boldsymbol{\Phi}(t_2, \tau)\boldsymbol{G}(\tau)\boldsymbol{u}(\tau)\mathrm{d}\tau, \tag{3.4}$$

and describes the evolution of the state $\boldsymbol{x}$ in the specified time interval $[t_1, t_2]$.

For time-invariant systems, the transition matrix is calculated using the matrix exponential as [36]

$$\boldsymbol{\Phi}(t_2, t_1) = \exp\left(\boldsymbol{F}(t_2 - t_1)\right). \tag{3.5}$$

The discrete and continuous system matrices can then be related by evaluating the state transition matrix at the defined time steps $t_1 = k$ and $t_2 = k+1$ resulting in [37, p. 45]

$$\boldsymbol{F}_k = \boldsymbol{\Phi}(k+1, k), \tag{3.6a}$$

$$\boldsymbol{G}_k = \boldsymbol{\Phi}(k+1, k)\boldsymbol{G}(k). \tag{3.6b}$$

3.2. Stochastic Properties and Definitions

The system state $\boldsymbol{x}$ and the input $\boldsymbol{u}$ do not necessarily need to be deterministic variables but can instead be exposed to a level of uncertainty. This holds for many applications, as signals are rarely noise free in real world applications [38]. Considering the uncertainty of signals and processes, the deterministic variables become *random variables* and the dynamics of a system need to be described as random or stochastic processes [39]. In the following, some properties of random variables and stochastic processes are introduced.

3.2.1. Scalar Random Variables

A random variable x maps the set of possible outcomes of an experiment to a range of real numbers [40, p. 53]. An example for this is the experiment of tossing a coin several times. Heads and tails of each coin toss are given the values -1 and 1 respectively and x denotes the sum of the outcome of all tosses. The *expected value* – also called *mean* – of such a random variable is denoted as $E(x)$ or $\bar{x}$. For the coin toss experiment it would be 0, which is also referred to as *zero-mean*.

The *variance* is defined as [40, p. 56]

$$\begin{aligned}\sigma_x^2 &= E[(x - \bar{x})^2] \\ &= E(x^2) - 2E(x)\bar{x} + \bar{x}^2 \\ &= E(x^2) - \bar{x}^2\end{aligned} \tag{3.7}$$

which is a measure by how much the actual x will vary from its expected value $\bar{x}$.

If several scalar random variables exist, e. g. x and y, their relation can be described using the following properties:

$$\begin{aligned}C_{xy} &= E[(x - \bar{x})(y - \bar{y})] \\ &= E(xy) - \bar{x}\bar{y}\end{aligned} \tag{3.8}$$

is the *covariance* and

$$R_{xy} = E(xy) \tag{3.9}$$

is the *correlation* of the two random variables [40, p. 62]. Both covariance and correlation are a way to describe the joint variability of to random variables. If an increase in one variable very likely introduces an increase in the other variable, the correlation will produce a high value and the random variables can be considered correlated. The covariance is similar to the correlation, but describes how the deviation of the mean of two random variables are correlated.

3.2.2. Vector-Valued Random Variables and Stochastic Processes

Random variables do not necessarily need to be scalar quantities. Consider two column-vector random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ with n and m elements respectively. Their stochastic properties can now be described analogous to the scalar case, using the following matrix-quantities [40, p. 66]:

- correlation:

$$\begin{aligned}\boldsymbol{R}_{xy} &= E(\boldsymbol{x}\boldsymbol{y}^T) \\ &= \begin{bmatrix} E(x_1y_1) & \cdots & E(x_1y_m) \\ \vdots & & \vdots \\ E(x_ny_1) & \cdots & E(x_ny_m) \end{bmatrix},\end{aligned} \tag{3.10}$$

- covariance

$$\begin{aligned}\boldsymbol{C}_{xy} &= E[(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{y} - \bar{\boldsymbol{y}})^T] && \text{(3.11a)} \\ &= E(\boldsymbol{x}\boldsymbol{y}^T) - \bar{\boldsymbol{x}}\bar{\boldsymbol{y}}^T \\ &= \boldsymbol{R}_{xy} - \bar{\boldsymbol{x}}\bar{\boldsymbol{y}}^T, && \text{(3.11b)}\end{aligned}$$

- autocorrelation

$$\boldsymbol{R}_{xx} = E(\boldsymbol{x}\boldsymbol{x}^T), \tag{3.12}$$

- autocovariance

$$\boldsymbol{C}_{xx} = E[(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^T] \tag{3.13}$$

with $\boldsymbol{R}_{xx}$ and $\boldsymbol{C}_{xx}$ always being symmetric and positive semidefinite matrices.

If a random variable changes with time t, it is called a *random process* or *stochastic process* [40, p. 68]. Evaluating a vector-valued stochastic process at two different times t_1 and t_2 can be treated as evaluating two random variables $\boldsymbol{x}(t_1)$ and $\boldsymbol{x}(t_2)$.

Additionally, the following general relation holds between covariance and correlation [37, p. 181]:

$$\boldsymbol{C}_{xy}(t_1, t_2) = \boldsymbol{R}_{xy}(t_1, t_2) - E(\boldsymbol{x}(t_1))E(\boldsymbol{y}^T(t_2)). \tag{3.14}$$

Note that for zero-mean random variables, both correlation and covariance are equal.

3.2.3. Normal Processes and White Noise

Many noisy processes can be modeled in such way, that the statistical calculations simplify. For this, the noise is described as normal process – also called Gaussian process – or even as white noise.

In normal processes, the outcome $\boldsymbol{x}$ is *normal distributed* or *gaussian distributed* around its mean value (see [40, p. 58] for details). A process is additionally called white, if the values $\boldsymbol{x}(t)$ are independent between different time steps. This can be seen with the covariance matrix for a white process, for example for the time discrete case [37, p. 194]

$$\boldsymbol{C}_{xx} = E\left[(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k)(\boldsymbol{x}_j - \bar{\boldsymbol{x}}_j)^T\right] = \boldsymbol{Q}(k)\delta_{kj} \tag{3.15}$$

with the Kronecker-delta

$$\delta_{jl} = \begin{cases} 1 & \text{for } j = k, \\ 0 & \text{else.} \end{cases} \tag{3.16}$$

The independence allows to write the covariance for each time step individually, which is denoted with $\boldsymbol{Q}(k)$.

3.2.4. Stochastic Properties in Dynamic Systems

In dynamic systems, the inputs and measurements are exposed to noise, which in turn influence the state of the system.

Generally, the propagation of the covariance in matrix-vector equations can be calculated: Consider a linear system of equations, for example the measurement equation from (3.2)

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} \tag{3.17}$$

with a given mean and covariance for $\boldsymbol{x}$ and the measurement matrix $\boldsymbol{H}$. The mean for $\boldsymbol{y}$ is calculated as

$$E(\boldsymbol{y}) = \boldsymbol{H}\bar{\boldsymbol{x}} \tag{3.18}$$

and the covariance as

$$\begin{aligned} \boldsymbol{C}_{yy} =& E[(\boldsymbol{y} - E(\boldsymbol{y}))(\boldsymbol{y} - E(\boldsymbol{y}))^T] \\ =& E[(\boldsymbol{H}\boldsymbol{x} - \boldsymbol{H}\bar{\boldsymbol{x}})(\boldsymbol{H}\boldsymbol{x} - \boldsymbol{H}\bar{\boldsymbol{x}})^T] \\ =& \boldsymbol{H}\,\boldsymbol{C}_{xx}\boldsymbol{H}^T. \end{aligned} \tag{3.19}$$

For dynamic systems of random processes, a similar relation holds. If the uncertainty of dynamic systems is considered, (3.3) is rewritten as [40, p. 124]

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k \boldsymbol{x}_k + \boldsymbol{G}_k \boldsymbol{u}_k + \boldsymbol{w}_k, \tag{3.20a}$$
$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{v}_k, \tag{3.20b}$$

with the deterministic input $\boldsymbol{u}$, the process noise $\boldsymbol{w}$ and the measurement noise $\boldsymbol{v}$, with the noises both being Gaussian zero-mean white noise. Additionally, the process noise covariance matrix $\boldsymbol{Q}$ and the measurement noise covariance matrix $\boldsymbol{R}$ are given.

The covariance matrix $\boldsymbol{P}$ of the state $\boldsymbol{x}$ can then be computed as [40, p. 108]

$$\boldsymbol{P}_k = \boldsymbol{F}_{k-1} \boldsymbol{P}_{k-1} \boldsymbol{F}_{k-1}^T + \boldsymbol{Q}_{k-1}. \tag{3.21}$$

Considering uncertainties and the discrete formulation, the nonlinear system (3.1) is also rewritten as

$$\boldsymbol{x}_k = \boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{w}_{k-1}), \tag{3.22a}$$
$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k, \boldsymbol{v}_k). \tag{3.22b}$$

3.3. Measurement of Random Variables and Over-Constrained Systems

The noise in a dynamic systems makes the use of special measurement techniques for the system output necessary. To obtain reasonable values from the noise-corrupted system observation, general techniques for the measurement of random variables can be used.

This section discusses the measurement of quantities. For the evaluation of sensor information the measurement noise needs to be taken into account. Additionally, several sensors can be available to measure the same quantity, thus resulting in an overdetermined measurement equation.

3.3.1. Weighted Generalized Inverse

Consider a linear system of equations

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x}, \tag{3.23}$$

with $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^m$ and $\boldsymbol{H} \in \mathbb{R}^{m \times n}$. Hereby, $\boldsymbol{y}$ can be seen as the collection of m measurements and $\boldsymbol{x}$ is the vector of n system variables. The task is to determine $\boldsymbol{x}$ using the collected measurements $\boldsymbol{y}$.

If $\boldsymbol{H}$ is a regular matrix ($n = m$ and rank($\boldsymbol{H}$) $= n$), the vector $\boldsymbol{x}$ can be calculated by

$$\boldsymbol{x} = \boldsymbol{H}^{-1}\boldsymbol{y}, \tag{3.24}$$

where $\boldsymbol{H}^{-1}$ denotes the inverse of matrix $\boldsymbol{H}$. For all other cases, no inverse $\boldsymbol{H}^{-1}$ exists - the linear system of equations (3.23) can be *overdetermined* for $n < m$ or *underdetermined* for $n > m$, or more generally, $\boldsymbol{H}$ doesn't have full rank. However, a solution can still be found using the concept of *weighted generalized inverses* [41, p. 117ff]. An overview on this topic can be found in [42], where the weighted generalized inverse $\boldsymbol{H}^{\#}$ is given as

$$\boldsymbol{H}^{\#} := \boldsymbol{M}_{\boldsymbol{x}}^{-1}\boldsymbol{G}^T\left(\boldsymbol{G}\boldsymbol{M}_{\boldsymbol{x}}^{-1}\boldsymbol{G}^T\right)^{-1}\left(\boldsymbol{F}^T\boldsymbol{M}_{\boldsymbol{y}}\,\boldsymbol{F}\right)^{-1}\boldsymbol{F}^T\boldsymbol{M}_{\boldsymbol{y}}, \tag{3.25}$$

with the weights $\boldsymbol{M}_{\boldsymbol{x}}$ and $\boldsymbol{M}_{\boldsymbol{y}}$, and the matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ as result of the *full rank decomposition*

$$\boldsymbol{H} = \boldsymbol{F}\boldsymbol{G} \qquad \text{with} \qquad \operatorname{rank}(\boldsymbol{H}) = \text{row-rank}(\boldsymbol{G}) = \text{column-rank}(\boldsymbol{F}). \tag{3.26}$$

The matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ can be determined via a generalized singular value decomposition as [43]

$$\boldsymbol{H} = \boldsymbol{F}\boldsymbol{G} = \boldsymbol{U}\left(\boldsymbol{\Sigma}\boldsymbol{V}^*\right). \tag{3.27}$$

Using $\boldsymbol{H}^{\#}$ allows to find the solution $\hat{\boldsymbol{x}}$ which "is the minimum $\boldsymbol{M}_{\boldsymbol{x}}$-norm solution that generates the $\boldsymbol{M}_{\boldsymbol{y}}$-least-squares error" [42]:

$$\hat{\boldsymbol{x}} = \boldsymbol{H}^{\#}\boldsymbol{y}. \tag{3.28}$$

This solution minimizes the following norms [42]:

$$J_1 = \hat{\boldsymbol{x}}^T\boldsymbol{M}_{\boldsymbol{x}}\hat{\boldsymbol{x}}, \tag{3.29a}$$

$$J_2 = \left(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}\right)^T\boldsymbol{M}_{\boldsymbol{y}}\left(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}\right). \tag{3.29b}$$

Special cases: For the cases that $\boldsymbol{H}$ has certain properties, (3.25) can be simplified [42]:

a) $\boldsymbol{H}$ has full row-rank. This renders $\boldsymbol{H}^{\#}$ invariant to the weight $\boldsymbol{M}_{\boldsymbol{y}}$ and results in

$$\boldsymbol{H}^{\#} = \boldsymbol{M}_{\boldsymbol{x}}^{-1}\boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{M}_{\boldsymbol{x}}^{-1}\boldsymbol{H}^T\right)^{-1}, \tag{3.30}$$

b) $\boldsymbol{H}$ has full column-rank. This renders $\boldsymbol{H}^{\#}$ invariant to the weight $\boldsymbol{M}_{\boldsymbol{x}}$ and only allows to choose the weights for the $\boldsymbol{M}_{\boldsymbol{y}}$-least-squares error:

$$\boldsymbol{H}^{\#} = \left(\boldsymbol{H}^T\boldsymbol{M}_{\boldsymbol{y}}\,\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{M}_{\boldsymbol{y}}, \tag{3.31}$$

The *Moore-Penrose pseudoinverse* $\boldsymbol{H}^{+}$ considers the special case $\boldsymbol{M}_{\boldsymbol{x}} = \boldsymbol{M}_{\boldsymbol{y}} = \boldsymbol{I}$, resulting in

$$\boldsymbol{H}^{+} = \boldsymbol{G}^T\left(\boldsymbol{G}\boldsymbol{G}^T\right)^{-1}\left(\boldsymbol{F}^T\,\boldsymbol{F}\right)^{-1}\boldsymbol{F}^T. \tag{3.32}$$

This can additionally be combined with the special cases a) and b) yielding the *right pseudoinverse*

$$\boldsymbol{H}^{+} = \boldsymbol{H}^{T}\left(\boldsymbol{H}\boldsymbol{H}^{T}\right)^{-1} \qquad \text{with} \qquad \boldsymbol{H}\boldsymbol{H}^{+} = \boldsymbol{I} \tag{3.33}$$

and the *left pseudoinverse*

$$\boldsymbol{H}^{+} = \left(\boldsymbol{H}^{T}\,\boldsymbol{H}\right)^{-1}\boldsymbol{H}^{T} \qquad \text{with} \qquad \boldsymbol{H}^{+}\boldsymbol{H} = \boldsymbol{I} \tag{3.34}$$

respectively.

In practice, the weight can be applied the following: If the solution vector $\hat{\boldsymbol{x}}$ consists of elements with different units, it might make little sense to minimize the euklidean norm $\|\hat{\boldsymbol{x}}\|_2$ but rather use $\boldsymbol{M}_x$ to adjust for the units. If there is more confidence in one measurement than in others, it can be given higher importance by increasing the respective value in $\boldsymbol{M}_y$ [40, p. 82]. Recall that both weights are only applicable if $\boldsymbol{H}$ does not have full column- or row-rank respectively. The generalized inverse using a specific weight $\boldsymbol{M}$ is denoted as $\boldsymbol{H}^{M+}$.

3.3.2. Least Squares Estimation

Section 3.3.1 introduced the mathematics behind the weighted generalized inverses providing tools to solve linear systems of equations. These tools are now applied to the task of *least squares estimation*.

Consider a linear system of equations

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{v}, \tag{3.35}$$

where $\boldsymbol{y}$ is a set of m measurements of the n system variables in $\boldsymbol{x}$ with the additional influence of measurement noise $\boldsymbol{v}$. Assume that the number of measurements is higher than the number of system variables, thus $m > n$. The task is to find an estimate $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$, which provides the *optimal* results. Such an estimate $\hat{\boldsymbol{x}}$ can be found following [40, p. 80f]:

Define $\tilde{\boldsymbol{x}} := \boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}$ as the measurement residual (error) and a cost function

$$J_2 = \tilde{x}_1^2 + \tilde{x}_2^2 + ... + \tilde{x}_m^2 = (\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}})^T(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}})\,. \tag{3.36}$$

The task to find the optimal estimate $\hat{\boldsymbol{x}}$ is the same as minimizing J_2, which is done by computing the partial derivative

$$\frac{\partial J_2}{\partial \hat{\boldsymbol{x}}} = -2\boldsymbol{y}^T\boldsymbol{H} + 2\hat{\boldsymbol{x}}^T\boldsymbol{H}^T\boldsymbol{H} \stackrel{!}{=} \boldsymbol{0}, \tag{3.37}$$

resulting in the following expression for $\hat{\boldsymbol{x}}$:

$$\hat{\boldsymbol{x}} = \left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{y} = \boldsymbol{H}^{+}\boldsymbol{y}, \tag{3.38}$$

which is the left (Moore-Penrose) pseudo-inverse (3.34).

If information on the noise is available, the least squares estimation can be adjusted accordingly. Assume that the noise $\boldsymbol{v}$ is white noise with zero-mean

$$E(v_i) = 0 \tag{3.39}$$

and the measurement covariance matrix is known as

$$\boldsymbol{R} = E(\boldsymbol{v}\boldsymbol{v}^T) = \mathrm{diag}(\sigma_1^2, ... \sigma_m^2). \tag{3.40}$$

The cost function is redefined as [40, p. 82f]

$$J_2 = \tilde{x}_1^2/\sigma_1^2 + ... + \tilde{x}_m^2/\sigma_m^2 = (\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}})^T \boldsymbol{R}^{-1} (\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}), \tag{3.41}$$

which changes (3.38) to

$$\hat{\boldsymbol{x}} = \left(\boldsymbol{H}^T \boldsymbol{R}^{-1} \boldsymbol{H}\right)^{-1} \boldsymbol{H}^T \boldsymbol{R}^{-1} \boldsymbol{y} = \boldsymbol{H}^{\boldsymbol{R}+} \boldsymbol{y}, \tag{3.42}$$

where $\boldsymbol{H}^{\boldsymbol{R}+}$ is the weighted generalized inverse (3.31) with the weight $\boldsymbol{M_y} = \boldsymbol{R}^{-1}$.

Recursive computation of the estimate: The recursive computation of the least squares estimate can be used, if the estimate has to be updated continuously to take new measurements into account [40, p. 84].

Assume that the estimate $\hat{\boldsymbol{x}}$ was calculated for $k-1$ measurements. If a new measurement $\boldsymbol{y}_k$ of the system arrives, the estimate $\hat{\boldsymbol{x}}_k$ can be computed using a linear recursive estimator with the new measurement and the previous estimate [40, p. 84f]:

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{v}_k \tag{3.43a}$$
$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k-1} + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_{k-1}) \tag{3.43b}$$

The gain matrix $\boldsymbol{K}_k$ is determined by minimizing the cost function J_k, which is set as the sum of variances of the estimation error $\tilde{\boldsymbol{x}}_k = \boldsymbol{x} - \hat{\boldsymbol{x}}_k$

$$\begin{aligned} J_k &= E(\tilde{x}_{1,k}^2 + ... + \tilde{x}_{n,k}^2) \\ &= E(\tilde{\boldsymbol{x}}_k^T \tilde{\boldsymbol{x}}_k) \\ &= \mathrm{trace}(\boldsymbol{P}_k), \end{aligned} \tag{3.44}$$

where $\boldsymbol{P}_k = E(\tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_k^T)$ is the error covariance matrix. The gain matrix $\boldsymbol{K}_k$ is then obtained by inserting (3.43) into the cost function and minimizing it (see (3.46a) and [40, p. 84f]).

In summary, the equations for the recursive least squares estimation are given as [40, p. 86]:

- Initialization:

$$\hat{\boldsymbol{x}}_0 = E(\boldsymbol{x}), \tag{3.45a}$$
$$\boldsymbol{P}_0 = E[(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^T]. \tag{3.45b}$$

- For all time steps $k = 1, 2, ...$, first the measurement $\boldsymbol{y}_k$ is obtained and then the gain, the estimate, and the estimation-error covariance are updated as follows:

$$\boldsymbol{K}_k = \boldsymbol{P}_{k-1}\boldsymbol{H}_k^T(\boldsymbol{H}_k\boldsymbol{P}_{k-1}\boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1}, \tag{3.46a}$$
$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k-1} + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_{k-1}), \tag{3.46b}$$
$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_{k-1}(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)^T + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^T. \tag{3.46c}$$

3.4. Kalman Filter

The concept of the recursive weighted least squares estimation can be applied to a dynamic system, where the system dynamics are additionally considered. The Kalman filter yields the expected value (mean) as estimate $\hat{\boldsymbol{x}}$ of the state $\boldsymbol{x}$ [40, p. 124], taking into account both the measurements and the system dynamics.

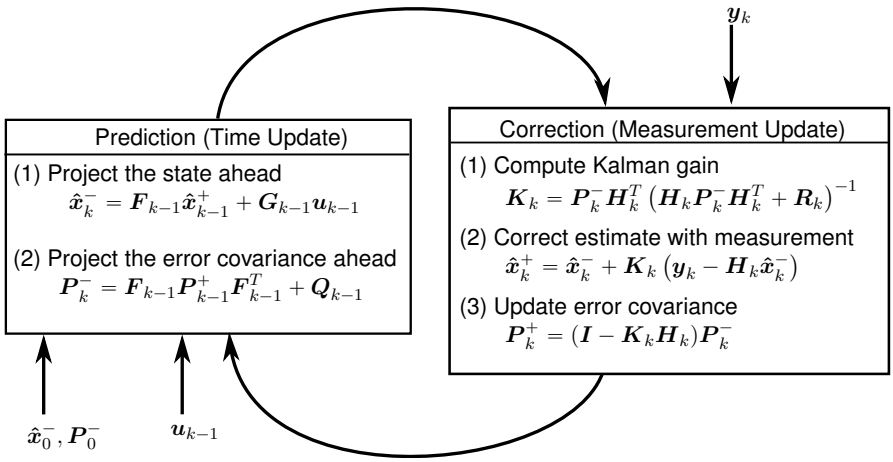

fig. 3.1.: The concept of Kalman filtering with the prediction and correction step. Adapted from [44].

Figure 3.1 illustrates the main concept of the Kalman filter. Instead of only measuring the system state at each time step – as it is done for the recursive least squares estimation – additionally, the system dynamics are used to calculate intermediate steps between the measurements. The measurement steps are therefore referred to as *correction* of the *prediction*, which is based on the system dynamics.

The description for the discrete-time Kalman filter is given in [40, p. 124ff]:

A linear time-discrete system (3.20) is assumed and the processes noise $\boldsymbol{w}_k$ and the measurement noise $\boldsymbol{v}_k$ are zero-mean, white, and uncorrelated:

$$E(\boldsymbol{w}_k) = \boldsymbol{0}, \tag{3.47a}$$
$$E(\boldsymbol{v}_k) = \boldsymbol{0}, \tag{3.47b}$$
$$E(\boldsymbol{w}_k \boldsymbol{w}_j^T) = \boldsymbol{Q}_k \delta_{k-j}, \tag{3.47c}$$
$$E(\boldsymbol{v}_k \boldsymbol{v}_j^T) = \boldsymbol{R}_k \delta_{k-j}, \tag{3.47d}$$
$$E(\boldsymbol{v}_k \boldsymbol{w}_j^T) = \boldsymbol{0}, \tag{3.47e}$$

with $j = 1, .., k$. Note the assumption that the process and measurement noises at time step k do not depend on previous time steps $j < k$, which is described by the Kronecker delta δ_{k-j}.

Recalling the recursive least squares estimation (3.46), one can see that for each time step k, values from the previous time step $k-1$ were used. The Kalman filter can be described as a recursive least squares estimation incorporating the system dynamics in a way such that intermediate predictions from the dynamic model are used instead of values from the previous time step $k-1$. Therefore, each time step can be further divided into a *prediction* part and an *correction* part [44].

The prediction estimate $\hat{\boldsymbol{x}}_k^-$ is the predicted future system state using the system dynamics (3.20) and the system state from the previous time step [40, p. 125]. Additionally, the prediction covariance matrix $\tilde{\boldsymbol{P}}_k^-$ is calculated using (3.21).

The correction estimate $\hat{\boldsymbol{x}}_k^+$ is the predicted estimate corrected by the performed measurements $\boldsymbol{y}_k$, where the recursive weighted least squares estimation is used to incorporate the measurements. Here, the equations (3.46) are used and the variables $\hat{\boldsymbol{x}}_k$, $\hat{\boldsymbol{x}}_{k-1}$, $\boldsymbol{P}_k$, and $\boldsymbol{P}_{k-1}$ are replaced with $\hat{\boldsymbol{x}}_k^+$, $\hat{\boldsymbol{x}}_k^-$, $\boldsymbol{P}_k^+$ and $\boldsymbol{P}_k^-$ respectively (see [40, p. 126ff]).

All of this results in the equations for the Kalman filter [40, p. 128f]:

- Initialization:

$$\hat{\boldsymbol{x}}_0^+ = E\left(\boldsymbol{x}_0\right), \tag{3.48a}$$
$$\boldsymbol{P}_0^+ = E\left((\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)^T\right). \tag{3.48b}$$

- Prediction at time-step k:

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{F}_{k-1}\hat{\boldsymbol{x}}_{k-1}^+ + \boldsymbol{G}_{k-1}\boldsymbol{u}_{k-1}, \tag{3.49a}$$
$$\boldsymbol{P}_k^- = \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1}^+\boldsymbol{F}_{k-1}^T + \boldsymbol{Q}_{k-1}, \tag{3.49b}$$

- Correction at time step k:

$$\boldsymbol{K}_k = \boldsymbol{P}_k^-\boldsymbol{H}_k^T\left(\boldsymbol{H}_k\boldsymbol{P}_k^-\boldsymbol{H}_k^T + \boldsymbol{R}_k\right)^{-1}, \tag{3.50a}$$
$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_k^-\right), \tag{3.50b}$$
$$\begin{aligned}\boldsymbol{P}_k^+ &= \left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k\right)\boldsymbol{P}_k^-\left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k\right)^T + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^T \\ &= \left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k\right)\boldsymbol{P}_k^-.\end{aligned} \tag{3.50c}$$

The derivation of the simplified form of (3.50c) can be found in [38, p. 137] and consists of several steps including the substitution of $\boldsymbol{K}$ with (3.50a).

These are the main general equations of the Kalman filter. A multitude of variations of these equations exist for different application purposes (see e. g. [40] or [38] for an overview). For this thesis, two further aspects on Kalman filtering need to be addressed:

Extended Kalman Filter: One special form of Kalman filter needs to be discussed in this section as it is used for the sensor fusion for pose estimation: The extended Kalman filter. The details on the fusion EKF for the LRU are presented in [35], in the following only the basic concept is outlined.

The previously introduced Kalman filter is only suited for linear systems. To use the Kalman filter for nonlinear system dynamics (see (3.22)), changes need to be applied.

For the EKF, the nonlinear system is linearized around the Kalman filter estimate and the linearized system is in turn used to find the Kalman filter estimate of the next time step [40, p. 400]. Therefore, the Kalman filter equations are changed to [40, p. 409]:

- Linearization of system dynamics and prediction step:

$$\boldsymbol{F}_{k-1} = \left.\frac{\partial \boldsymbol{f}_{k-1}}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k-1}^{+}}, \tag{3.51a}$$

$$\boldsymbol{L}_{k-1} = \left.\frac{\partial \boldsymbol{f}_{k-1}}{\partial \boldsymbol{w}}\right|_{\hat{\boldsymbol{x}}_{k-1}^{+}}, \tag{3.51b}$$

$$\hat{\boldsymbol{x}}_{k}^{-} = \boldsymbol{f}_{k-1}\left(\hat{\boldsymbol{x}}_{k-1}^{+}, \boldsymbol{u}_{k-1}, \boldsymbol{0}\right), \tag{3.51c}$$

$$\boldsymbol{P}_{k}^{-} = \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1}^{+}\boldsymbol{F}_{k-1}^{T} + \boldsymbol{L}_{k-1}\boldsymbol{Q}_{k-1}\boldsymbol{L}_{k-1}^{T}, \tag{3.51d}$$

- Linearization of measurement equation and correction step:

$$\boldsymbol{H}_{k} = \left.\frac{\partial \boldsymbol{h}_{k}}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k}^{-}}, \tag{3.52a}$$

$$\boldsymbol{M}_{k} = \left.\frac{\partial \boldsymbol{h}_{k}}{\partial \boldsymbol{v}}\right|_{\hat{\boldsymbol{x}}_{k}^{-}} \tag{3.52b}$$

$$\boldsymbol{K}_{k} = \boldsymbol{P}_{k}^{-}\boldsymbol{H}_{k}^{T}\left(\boldsymbol{H}_{k}\boldsymbol{P}_{k}^{-}\boldsymbol{H}_{k}^{T} + \boldsymbol{M}_{k}\boldsymbol{R}_{k}\boldsymbol{M}_{k}^{T}\right)^{-1}, \tag{3.52c}$$

$$\hat{\boldsymbol{x}}_{k}^{+} = \hat{\boldsymbol{x}}_{k}^{-} + \boldsymbol{K}_{k}\left(\boldsymbol{y}_{k} - \boldsymbol{h}_{k}(\hat{\boldsymbol{x}}_{k}^{-}, \boldsymbol{0})\right), \tag{3.52d}$$

$$\boldsymbol{P}_{k}^{+} = (\boldsymbol{I} - \boldsymbol{K}_{k}\boldsymbol{H}_{k})\boldsymbol{P}_{k}^{-}. \tag{3.52e}$$

An overview on the use of an EKF for the localization of mobile robots can be found in [45, p. 191ff].

Sensor Fusion: The other important aspects addresses the incorporation of redundant measurements obtained by different sensors into the filtering process. Mobile robots generally have a multitude of sensors available to perceive the environment. Examples are camera systems, wheel odometry, or inertial sensors. These are incorporated into the filtering process by weighting the sensor measurements accordingly to their uncertainties [45, p. 149ff]. As the Kalman filter considers stochastic rather than deterministic processes, the measurement equation (3.20b) already includes the information on uncertainty in the measurement covariance $\boldsymbol{R}$. Additional sensor inputs result in additional entries in $\boldsymbol{y}$ and $\boldsymbol{R}$. The measurement covariance determined by modeling the stochastic aspects of the sensor signal and accounts for sensor inaccuracies [45, p. 150].

4. Wheel Odometry

Wheel odometry is the pose estimation of an vehicle done by evaluating the sensor information of the wheels and the other body joints. There, the sensor information is used to determine the body velocity which in return is integrated to obtain the rover pose.

As first step towards the computation of the wheel odometry for the LRU, a kinematic model of the rover is required. The modeling is done by assigning several local coordinate frames to the body links of the rover and determining the transformations between them. The body links are the main body, the front and rear bogies, and the wheels.

Generally, the description of positions and directions is done using homogeneous coordinates. These read $[x_i, y_i, w]^T$ in two and $[x_i, y_i, z_i, w]^T$ in three dimensions respectively, where the position is expressed relative to a frame i. To describe a point in space, w is set to 1, for a direction in space it reads 0.

Any vector of homogenous coordinates can be described in different frames. The transformation of a vector $\boldsymbol{r}$ between two arbitrary frames A and B can be described by a homogeneous transformation

$$_{\mathrm{A}}\boldsymbol{r} = \boldsymbol{H}_{\mathrm{AB}\,\mathrm{B}}\boldsymbol{r}, \tag{4.1}$$

where $\boldsymbol{H}_{\mathrm{AB}}$ resembles the transformation matrix, that transforms the vector $\boldsymbol{r}$ from the frame B into the frame A.

The homogeneous transformation matrix includes a translation part $\boldsymbol{p}$ and a rotation $\boldsymbol{R}$ such that

$$\boldsymbol{H}_{\mathrm{AB}} = \begin{bmatrix} \boldsymbol{R}_{\mathrm{AB}} & \boldsymbol{p}_{\mathrm{AB}} \\ \boldsymbol{0} & 1 \end{bmatrix}. \tag{4.2}$$

Throughout this thesis, homogeneous transformations are used to describe the positions and directions of the rover and its sub-components. To calculate linear and angular velocities of these components, the rotational and translational parts of the transformation are used individually.

The following relation holds for reversing transformations [46, p. 37]:

$$\boldsymbol{H}_{\mathrm{BA}} = \boldsymbol{H}_{\mathrm{AB}}^{-1} = \begin{bmatrix} \boldsymbol{R}_{\mathrm{AB}}^T & -\boldsymbol{R}_{\mathrm{AB}}^T\boldsymbol{p}_{\mathrm{AB}} \\ \boldsymbol{0} & 1 \end{bmatrix}. \tag{4.3}$$

A transformation including several steps between various frames can be calculated by multiplying the matrices [19]

$$\boldsymbol{H}_{\mathrm{AC}} = \boldsymbol{H}_{\mathrm{AB}}\,\boldsymbol{H}_{\mathrm{BC}}. \tag{4.4}$$

4.1. 2D Kinematics

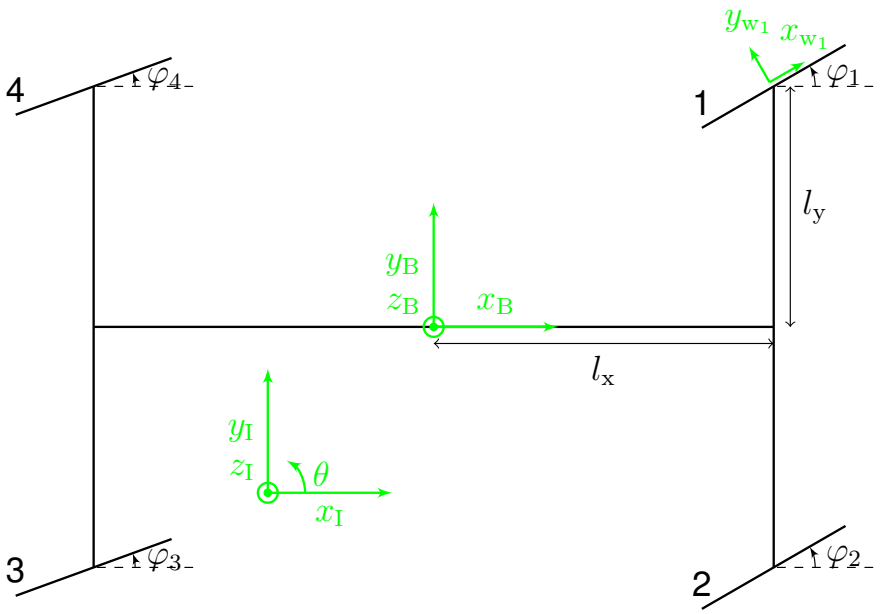

fig. 4.1.: 2D LRU kinematics - geometry and conventions.

The LRU kinematics can be modeled in two and three dimensions, with the two-dimensional case as simplification, assuming planar terrain. The frames for the two-dimensional case are displayed in figure 4.1. The indices I and B denote the inertial and the body-fixed frames respectively. The frames of the wheels are denoted as w_i with $i = 1, ..., 4$ and are exemplary depicted for wheel 1.

The pose $\boldsymbol{\xi}$ of a rigid body in a two dimensional space can be described in terms of the three DOF of the body and reads for the LRU

$$\boldsymbol{\xi}_{\mathrm{I}} = [x, y, \theta]^T \tag{4.5}$$

expressed in the inertial frame. This representation is also referred to as the two-dimensional *spatial coordinates* of the rover body [46, p. 54].

For the two-dimensional case, the transformation matrix to relate the body frame B to the inertial frame I is defined as

$$\boldsymbol{H}_{\mathrm{IB}} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & x \\ \sin(\theta) & \cos(\theta) & y \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.6}$$

using the spatial coordinates $\boldsymbol{\xi}_I$. The transformation matrices between the body and the wheels w_i are defined as

$$\boldsymbol{H}_{\mathrm{Bw}_i} = \begin{bmatrix} \cos(\varphi_i) & -\sin(\varphi_i) & k_{\mathrm{front}}\, l_x \\ \sin(\varphi_i) & \cos(\varphi_i) & k_{\mathrm{left}}\, l_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4.7}$$

with

$$k_{\mathrm{front}} = \begin{cases} 1 & \text{for} \quad i = 1, 2, \\ -1 & \text{for} \quad i = 3, 4, \end{cases} \tag{4.8}$$

and

$$k_{\mathrm{left}} = \begin{cases} 1 & \text{for} \quad i = 1, 4, \\ -1 & \text{for} \quad i = 2, 3. \end{cases} \tag{4.9}$$

4.2. 3D Kinematics

Similar to the previously discussed 2D case, several frames are introduced for the three-dimensional kinematic model. The locations of the frames and the naming conventions are shown in figure 4.2.

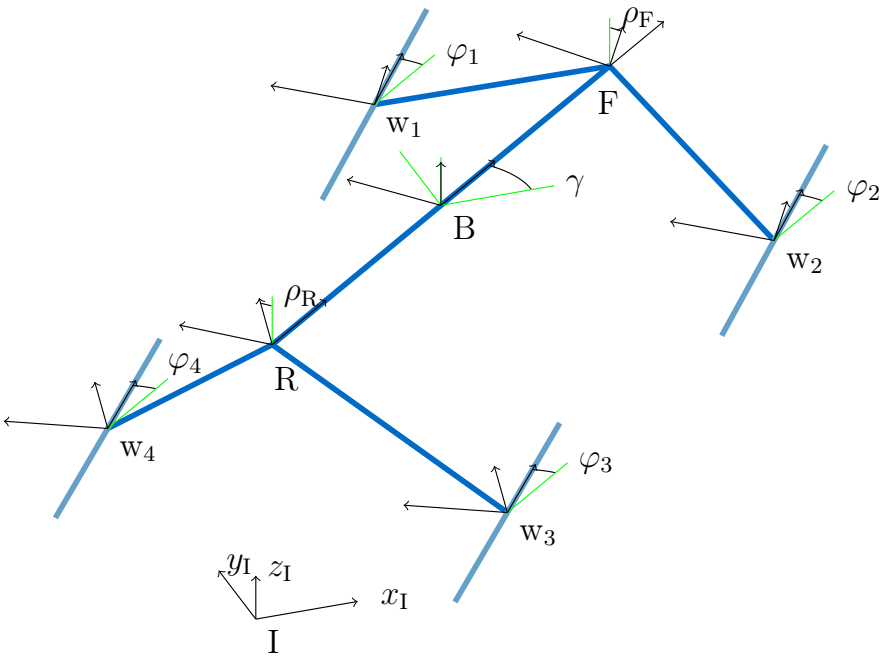

fig. 4.2.: 3D LRU kinematics - geometry and conventions.

The transformation between the inertial frame I and the body frame B is calculated using the *cardan-angle parameterization* (see [47, p. 374]). In this parameterization, three consecutive rotations are performed, rotating around the x, y,

and z axis of the respective intermediate frames by the roll angle α, the pitch β and the yaw γ. The resulting transformation matrix reads

$$\boldsymbol{H}_{\mathrm{IB}} = \begin{bmatrix} \mathrm{c}\beta\,\mathrm{c}\gamma & -\mathrm{c}\beta\,\mathrm{s}\gamma & \mathrm{s}\beta & {}_{\mathrm{I}}x \\ \mathrm{c}\alpha\,\mathrm{s}\gamma + \mathrm{s}\alpha\,\mathrm{s}\beta\,\mathrm{c}\gamma & \mathrm{c}\alpha\,\mathrm{c}\gamma - \mathrm{s}\alpha\,\mathrm{s}\beta\,\mathrm{s}\gamma & -\mathrm{s}\alpha\,\mathrm{c}\beta & {}_{\mathrm{I}}y \\ \mathrm{s}\alpha\,\mathrm{s}\gamma - \mathrm{c}\alpha\,\mathrm{s}\beta\,\mathrm{c}\gamma & \mathrm{s}\alpha\,\mathrm{c}\gamma + \mathrm{c}\alpha\,\mathrm{s}\beta\,\mathrm{s}\gamma & \mathrm{c}\alpha\,\mathrm{c}\beta & {}_{\mathrm{I}}z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.10}$$

with c and s denoting the cosine and sinus function of the respective angle. The scalars x, y, and z describe the position of the origin of frame B expressed in the inertial frame.

The suspensions of the two front and the two rear wheels are connected to the body center via *bogie joints*, which allow for a rotation along the x-axis of the body. The angles for the front and rear rotations are denoted as ρ_{F} and ρ_{R} and the corresponding transformation matrices between the links read

$$\boldsymbol{H}_{\mathrm{BF/BR}} = \begin{bmatrix} 1 & 0 & 0 & k_{\mathrm{front}} l_{\mathrm{x}} \\ 0 & \mathrm{c}\rho_{\mathrm{F/R}} & -\mathrm{s}\rho_{\mathrm{F/R}} & 0 \\ 0 & \mathrm{s}\rho_{\mathrm{F/R}} & \mathrm{c}\rho_{\mathrm{F/R}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.11}$$

with

$$k_{\mathrm{front}} = \begin{cases} 1 & \text{for the front bogie (wheels } i = 1, 2), \\ -1 & \text{for the rear bogie (wheels } i = 3, 4). \end{cases} \tag{4.12}$$

The next frames on the LRU are the axle frames. Their origin is identical to the origin of the wheel frames and is located in the center of the wheel hub, but the orientation is identical to the bogie frames. This results in the following transformations between the bogies and the four wheel axles a_i, which read

$$\boldsymbol{H}_{\mathrm{F/Ra}_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & k_{\mathrm{left}}\, l_{\mathrm{y}} \\ 0 & 0 & 1 & -l_{\mathrm{z}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.13}$$

with

$$k_{\mathrm{left}} = \begin{cases} 1 & \text{for} \quad i = 1, 4, \\ -1 & \text{for} \quad i = 2, 3. \end{cases} \tag{4.14}$$

Note that the frames on the wheel axles are defined with respect to the rigid body of the bogies and not on the wheels. The relationship between a frame on

the wheel axle and its respective frame on the wheels can then be expressed as

$$\boldsymbol{H}_{\mathrm{a}_i\mathrm{w}_i} = \begin{bmatrix} \mathrm{c}\varphi_i & -\mathrm{s}\varphi_i & 0 & 0 \\ \mathrm{s}\varphi_i & \mathrm{c}\varphi_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.15}$$

For the wheel odometry, the wheels are related to the main body of the rover by [19]

$$\boldsymbol{H}_{\mathrm{Bw}_i} = \boldsymbol{H}_{\mathrm{BF/R}}\, \boldsymbol{H}_{\mathrm{F/Ra}_i}\, \boldsymbol{H}_{\mathrm{a}_i\mathrm{w}_i}, \tag{4.16}$$

which results in following relation for the position $\boldsymbol{r}$ of a point p, both expressed in the wheel's frame and the body frame respectively:

$${}_{\mathrm{B}}\boldsymbol{r}_{\mathrm{p}} = \boldsymbol{H}_{\mathrm{Bw}_i\ \mathrm{w}_i}\boldsymbol{r}_{\mathrm{p}}. \tag{4.17}$$

The rotation matrices and translational transformations between the listed frames can be extracted from the homogeneous transformation according to equation (4.2).

The rover configuration can be expressed by a minimal set of coordinates, also called the vehicle state $[\boldsymbol{\zeta}, \rho_{\mathrm{F}}, \rho_{\mathrm{R}}]$, containing the rover pose $\boldsymbol{\zeta}$ and the bogie joint angles ρ. Following [21], the state is split up into free and fixed state variables. The fixed variables are predefined by the rover control and can be measured directly. These are the bogie angles ρ. The pose $\boldsymbol{\zeta}$ contains the six DOF to represent the position and orientation of the rover in three dimensional space. The pose described in the inertial frame using the cardan-angle parameterization and is defined as

$$\boldsymbol{\zeta} = [{}_{\mathrm{I}}x,\ {}_{\mathrm{I}}y,\ {}_{\mathrm{I}}z, \alpha, \beta, \gamma]^T. \tag{4.18}$$

The body velocity of the rover can be defined in two ways, using the angular rates of the cardan angles or alternatively using *spatial velocities*. The first definition uses the time derivatives of the cardan angles and is expressed with respect to the inertial frame. It reads

$$\dot{\boldsymbol{\zeta}}_{\mathrm{I}} = [{}_{\mathrm{I}}\dot{x},\ {}_{\mathrm{I}}\dot{y},\ {}_{\mathrm{I}}\dot{z}, \dot{\alpha}, \dot{\beta}, \dot{\gamma}]^T. \tag{4.19}$$

The body velocity can also be expressed via spatial velocities, which are made up of the linear and angular velocity components in the body frame resulting in [46, p. 54]

$$\dot{\boldsymbol{\zeta}}_{\mathrm{B}} = \hat{\boldsymbol{V}}_{\mathrm{B}} = \begin{bmatrix} \boldsymbol{v}_{\mathrm{B}} \\ \boldsymbol{\omega}_{\mathrm{B}} \end{bmatrix}. \tag{4.20}$$

The transformation of the velocities between the body frame and the inertial frame are outlined in section 4.3.1.

4.3. Body Velocity of the LRU

The main task of the wheel odometry is to determine the velocities $\dot{\zeta}_{\mathrm{B}}$ of the LRU-body and provide these to the fusion EKF, as it was outlined in section 2.3. This is done by mapping the speeds of the single wheels to the body.

Assume that the velocities

$$v_{\mathrm{wc}_i} = \omega_i r_i \tag{4.21}$$

of the wheel centers wc_i and the wheel contact angles ψ_i (depicted in figure 4.3) can be measured or estimated. This equation implicitly contains the ideal contact condition between the wheel rim and the terrain, as the rotational velocity of the wheels gets mapped as linear velocity onto the wheel center. The velocity of the wheel center can therefore be written as

$${}_{\mathrm{w}_i}\boldsymbol{v}_{\mathrm{wc}_i} = \begin{bmatrix} v_{\mathrm{wc}_i}\mathrm{c}\psi_i \\ 0 \\ v_{\mathrm{wc}_i}\mathrm{s}\psi_i \end{bmatrix}, \tag{4.22}$$

and is expressed in the wheel frame w_i.

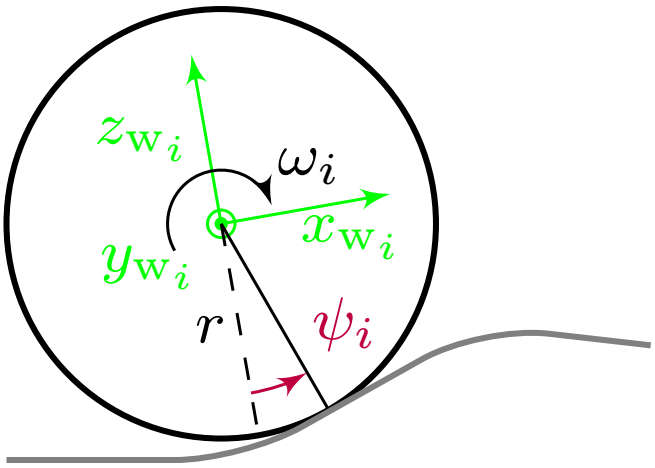

fig. 4.3.: Wheel contact angle ψ_i and the wheel frame of wheel i.

Using the information on the wheel speeds and the rover kinematics allows to calculate the vector $\dot{\zeta}$ as shown here:

To relate the measured wheel velocity to the body frame, a kinematic chain over all links of the LRU is introduced following [47, p. 144ff]: Generally, the velocity of a consecutive frame $j+1$ in the kinematic chain can be expressed in terms of the predecessor frame j and the relative velocity components between both frames. The angular velocities $\boldsymbol{\omega}$ can be related as

$${}_{j+1}\boldsymbol{\omega}_{j+1} = \boldsymbol{R}_{j+1,j}\ {}_{j}\boldsymbol{\omega}_{j} +_{j+1} \boldsymbol{e}_{j+1}\,\dot{\theta}_{j+1}. \tag{4.23}$$

This provides the rotational velocity of link $j+1$ expressed in the body fixed frame of link $j+1$, where the relative rotational velocity $\dot{\theta}_{j+1}$ at the intermediate joint around the rotation axis $\boldsymbol{e}_{j+1}$ is considered.

Analogously, the linear velocity components are computed as

$$ {}_{j+1}\boldsymbol{v}_{j+1} = \boldsymbol{R}_{j+1,j} \left({}_{j}\boldsymbol{v}_{j} + {}_{j}\boldsymbol{\omega}_{j} \times {}_{j}\boldsymbol{p}_{j+1} \right), \tag{4.24} $$

with the position $\boldsymbol{p}$ of joint $j+1$ expressed in frame j.

Applied to the LRU, the kinematic chain is started at the body frame B. In this frame, the angular velocity $\boldsymbol{\omega}_{\mathrm{B}}$ of the body is defined as

$$ {}_{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{B}} = {}_{\mathrm{B}}\left[\omega_x, \omega_y, \omega_z\right]^T. \tag{4.25} $$

The angular velocities of the next links, the front and rear bogies F and R are calculated as

$$ {}_{\mathrm{F/R}}\boldsymbol{\omega}_{\mathrm{F/R}} = \boldsymbol{R}_{\mathrm{F/R\,B}}\, {}_{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{B}} + {}_{\mathrm{F/R}}\boldsymbol{e}_x\, \dot{\rho}_{\mathrm{F/R}}, \tag{4.26} $$

with a relative rotational velocity $\dot{\rho}$ around the x-axes of the bogie frames.

For the wheels, the angular velocities are computed in the axle frames a_i and read

$$ {}_{\mathrm{a}_i}\boldsymbol{\omega}_{\mathrm{a}_i} = {}_{\mathrm{F/R}}\, \boldsymbol{\omega}_{\mathrm{F/R}}. \tag{4.27} $$

The linear velocities of the body frame are defined as

$$ {}_{\mathrm{B}}\boldsymbol{v}_{\mathrm{B}} = {}_{\mathrm{B}}\left[v_x, v_y, v_z\right]^T. \tag{4.28} $$

Following (4.24), the linear velocities of the other links are obtained as

$$ {}_{\mathrm{F/R}}\boldsymbol{v}_{\mathrm{F/R}} = \boldsymbol{R}_{\mathrm{F/R\,B}} \left({}_{\mathrm{B}}\boldsymbol{v}_{\mathrm{B}} + {}_{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{B}} \times {}_{\mathrm{B}}\boldsymbol{p}_{\mathrm{F/R}} \right) \tag{4.29} $$

and

$$ {}_{\mathrm{a}_i}\boldsymbol{v}_{\mathrm{a}_i} = \boldsymbol{R}_{\mathrm{a}_i\,\mathrm{F/R}} \left({}_{\mathrm{F/R}}\boldsymbol{v}_{\mathrm{F/R}} + {}_{\mathrm{F/R}}\boldsymbol{\omega}_{\mathrm{F/R}} \times {}_{\mathrm{F/R}}\boldsymbol{p}_{\mathrm{a}_i} \right). \tag{4.30} $$

So far, (4.30) allows to express the wheel velocities in terms of the body velocities. For the wheel odometry, this equation can be solved for the body velocities, which can then be computed using the measured wheel velocities. The wheel speeds are measured in the wheel frames and are converted to the axle frames by

$$ {}_{\mathrm{a}_i}\boldsymbol{v}_{\mathrm{a}_i} = \boldsymbol{R}_{\mathrm{a}_i\mathrm{w}_i}\, {}_{\mathrm{w}_i}\boldsymbol{v}_{\mathrm{w}_i} \tag{4.31} $$

The wheel velocities can be expressed in the body frame via

$$ {}_{\mathrm{B}}\boldsymbol{v}_{\mathrm{a}_i} = \boldsymbol{R}_{\mathrm{B\,F/R}} \boldsymbol{R}_{\mathrm{F/R\,a}_i}\, {}_{\mathrm{a}_i}\boldsymbol{v}_{\mathrm{a}_i}. \tag{4.32} $$

This equation contains the desired velocity components of the body frame, which are now extracted. Equation (4.32) is rewritten and the free and fixed derivatives of the state variables are separated [21]. For each wheel i, the equation reads

$$ {}_{\mathrm{B}}\boldsymbol{v}_{\mathrm{a}_i} = \boldsymbol{J}_{\mathrm{free}} \begin{bmatrix} {}_{\mathrm{B}}\boldsymbol{v}_{\mathrm{B}} \\ {}_{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{B}} \end{bmatrix} + \boldsymbol{J}_{\mathrm{fixed}} \dot{\rho}_{\mathrm{F/R}}, \tag{4.33} $$

where the respective free and fixed Jacobian matrices $\boldsymbol{J}_{\text{free}}$ and $\boldsymbol{J}_{\text{fixed}}$ are used. These Jacobian matrices allow to describe the wheel speeds in terms of the previously defined DOF of the system and are therefore referred to as *wheel Jacobian matrices* [19]. Equations (4.31), (4.32), and (4.20) are inserted and the equation is rearranged to

$$\boldsymbol{J}_{\text{free}}\dot{\boldsymbol{\zeta}}_{\text{B}} = \boldsymbol{R}_{\text{Bw}_i\ \text{w}_i}\boldsymbol{v}_{\text{w}_i} - \boldsymbol{J}_{\text{fixed}}\dot{\rho}_{\text{F/R}}. \tag{4.34}$$

To account for all four wheels, the resulting equations for $i = 1, ..., 4$ are stacked and solved for the body velocities:

$$\dot{\boldsymbol{\zeta}}_{\text{B}} = \begin{bmatrix} \boldsymbol{J}_{\text{free},1} \\ \boldsymbol{J}_{\text{free},2} \\ \boldsymbol{J}_{\text{free},3} \\ \boldsymbol{J}_{\text{free},4} \end{bmatrix}^{+} \left(\begin{bmatrix} \boldsymbol{R}_{\text{Bw}_1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_{\text{Bw}_2} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}_{\text{Bw}_3} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}_{\text{Bw}_4} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{\text{w}_1} \\ \boldsymbol{v}_{\text{w}_2} \\ \boldsymbol{v}_{\text{w}_3} \\ \boldsymbol{v}_{\text{w}_4} \end{bmatrix} - \begin{bmatrix} \boldsymbol{J}_{\text{fix},1}\,\dot{\rho}_F \\ \boldsymbol{J}_{\text{fix},2}\,\dot{\rho}_F \\ \boldsymbol{J}_{\text{fix},3}\,\dot{\rho}_R \\ \boldsymbol{J}_{\text{fix},4}\,\dot{\rho}_R \end{bmatrix} \right), \tag{4.35}$$

where the left Moore-Penrose pseudo inverse (3.34) is applied to the stacked free wheel Jacobian matrices, as the system is overdetermined. Note that the frame description via the left subscript is dropped for all frame velocities, if the velocities are expressed in their own frames. This is applies to all following velocities in this thesis and is used for simplification. The free wheel Jacobian matrix of a single wheel can be written as

$$\boldsymbol{J}_{\text{free},i} = \begin{bmatrix} 1 & 0 & 0 & 0 & \text{s}\rho\, k_{\text{left}} l_{\text{y}} - \text{c}\rho\, l_{\text{z}} & -\text{c}\rho\, k_{\text{left}} l_{\text{y}} - \text{s}\rho\, l_{\text{z}} \\ 0 & 1 & 0 & -\text{s}\rho\, k_{\text{left}} l_{\text{y}} + \text{c}\rho\, l_{\text{z}} & 0 & k_{\text{front}} l_{\text{x}} \\ 0 & 0 & 1 & \text{c}\rho\, k_{\text{left}} l_{\text{y}} + \text{s}\rho\, l_{\text{z}} & -k_{\text{front}} l_{\text{x}} & 0 \end{bmatrix} \tag{4.36}$$

and the fixed wheel Jacobian matrix as

$$\boldsymbol{J}_{\text{fix},i} = \begin{bmatrix} 0 \\ -\text{s}\rho\, k_{\text{left}} l_{\text{y}} + \text{c}\rho\, l_{\text{z}} \\ \text{c}\rho\, k_{\text{left}} l_{\text{y}} + \text{s}\rho\, l_{\text{z}} \end{bmatrix}. \tag{4.37}$$

4.3.1. Velocities Expressed in the Inertial Frame

The computed rover velocities can finally be expressed in the inertial frame by considering the cardan-angle-parameterization. The rotation matrix for the linear velocities is obtained from (4.10) and the angular velocities can be converted to the cardan angular velocities following [47, p. 143f] by

$$\boldsymbol{W}_{\text{IB}} = \frac{1}{\text{c}\beta} \begin{bmatrix} \text{c}\gamma & -\text{s}\gamma & 0 \\ \text{s}\gamma\,\text{c}\beta & \text{c}\gamma\,\text{c}\beta & 0 \\ -\text{c}\gamma\,\text{s}\beta & \text{s}\gamma\,\text{s}\beta & \text{c}\beta \end{bmatrix}. \tag{4.38}$$

Combined, the rover velocities can be calculated as

$$\dot{\boldsymbol{\zeta}}_{\mathrm{I}} = \begin{bmatrix} \boldsymbol{R}_{\mathrm{IB}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{W}_{\mathrm{IB}} \end{bmatrix} \dot{\boldsymbol{\zeta}}_{\mathrm{B}}. \tag{4.39}$$

4.3.2. Body Velocity with 2D Odometry

Analogue to the described 3D case, the 2D wheel Jacobians can be obtained. For this case, the calculation is simplified, as the bogie angles are not considered and the DOF are reduced to 3. The resulting wheel Jacobian for a single wheel in two dimensional space reads

$$\boldsymbol{J}_{2\mathrm{D},i} = \begin{bmatrix} 1 & 0 & -k_{\mathrm{left}} l_{\mathrm{y}} \\ 0 & 1 & k_{\mathrm{front}} l_{\mathrm{x}} \end{bmatrix}. \tag{4.40}$$

The body velocities $\dot{\boldsymbol{\xi}}_{\mathrm{B}}$ are obtained analogous to (4.35) and transformed to the inertial frame by

$$\dot{\boldsymbol{\xi}}_{I} = \begin{bmatrix} \boldsymbol{R}_{\mathrm{IB}} & \mathbf{0} \\ 0 & 1 \end{bmatrix} \dot{\boldsymbol{\xi}}_{\mathrm{B}}, \tag{4.41}$$

with $\boldsymbol{R}_{\mathrm{IB}}$ from (4.6).

4.3.3. Use of Wheel Odometry in the System Architecture

Recall that the body velocities provided by the wheel odometry are partly incorporated into the whole LRU system via the EKF. So far, the fusion filter only uses wheel odometry information on the linear velocity in x_{B} and y_{B} direction.

However, the presented wheel odometry is already capable to provide information on all three of the linear velocities and the yaw velocity. This framework therefore allows for future enhancements in the fusion filter.

The wheel odometry could additionally be augmented to compute the roll and pitch velocities. Due to the current set of sensors, the attitude angles of roll and pitch are not observable by the wheel odometry. However, a wheel contact angle estimation could be added to the system following for example [48]. Until such contact angle estimation is implemented, the fusion filter needs to obtain attitude information α and β by other sensors, namely the IMU.

The pose in the inertial frame is additionally computed – by wheel odometry alone – to allow for an usage for control purposes. The pitch and roll are hereby obtained from the EKF output.

Note that for $\rho \equiv 0$ and no information on roll and pitch, the two-dimensional wheel odometry is equivalent to the three-dimensional one.

4.4. Calibration

The wheel odometry relies on the exact information on geometry and precise measurements by the sensors. Errors in these will cause erroneous information about the rover velocity. The calibration of the rover allows to detect such errors and is therefore an important feature to improve wheel odometry. There exist many methods to do a detailed calibration as for example presented in [49], [24], or [25]. The first and usually most effective step for calibration is done manually.

This section lists all applied measures for calibration and then compares experimental results of the wheel odometry before and after the calibration.

4.4.1. Manual Calibration

During the process of manual calibration, the rover geometry was measured and corrected compared to the values provided by the data sheet [50].

The weight of the rover combined with its elastic components cause a different wheel base compared to the values specified in the data sheet, as the bogies and suspension experience an elastic deformation due to the rover mass. This deformation was measured at both LRUs, while they were fully equipped with all operational components like batteries or the robotic arm in the case of LRU-2. The measured distances between the centers of the wheels are listed in table 4.1.

tab. 4.1.: Adjusted geometry values of the LRUs: Distance between wheel centers.

Distance $[\mathrm{m}]$	Datasheet [50]	LRU-1	LRU-2
x-distance	0.840	0.860	0.864
y-distance	0.580	0.586	0.590

The nominal wheel radii $r = 0.125\,\mathrm{m}$ and the vertical height $l_z = 0.365\,\mathrm{m}$ [50] were confirmed.

Additionally to the wheel base, the steering angles where calibrated as well. The wheels where aligned parallel to the x-direction of the rover, which is equivalent to the initialization position $\varphi_i = 0$. The real potentiometer values at the steering angle $\varphi_i = 0$ where measured and used to correct the wheel-initialization-position. By this, a significant steering offset was detected and corrected afterwards, as it is shown on table 4.2. The range of potentiometer values over 360° is $0 - 10000$.

tab. 4.2.: Adjusted steering of the LRUs in degrees and potentimeter values.

	LRU-1 [°]	Init pre [−]	Init post [−]	LRU-2 [°]	Init pre [−]	Init post [−]
$\Delta\varphi_1$	0.74	6500	6370	-0.17	4560	4590
$\Delta\varphi_2$	0.23	2860	2900	0.37	4010	4075
$\Delta\varphi_3$	0.06	6160	6170	0.32	5610	5665
$\Delta\varphi_4$	0.00	5270	5270	-0.12	4940	4920

4.4.2. Selection of Sensor Signals

The next step of calibration considers the selection of signals. The sensors measuring the wheel speed provide measurements at a frequency of $1\,\mathrm{kHz}$ which experience high levels of noise. This noise can be reduced by computing the wheel speeds differently. The encoder position signals θ_i of the wheel motors are sampled at a frequency of $100\,\mathrm{Hz}$ instead and the difference between each sample is computed. The resulting angular velocities of the wheels are calculated as

$$\omega_{i,k} = \frac{\theta_{i,k} - \theta_{i,k-1}}{T}, \tag{4.42}$$

with the sampling period $T = 0.01\,\mathrm{s}$. A comparison between the two methods can be seen in figure 4.4, which illustrates the difference in noise.

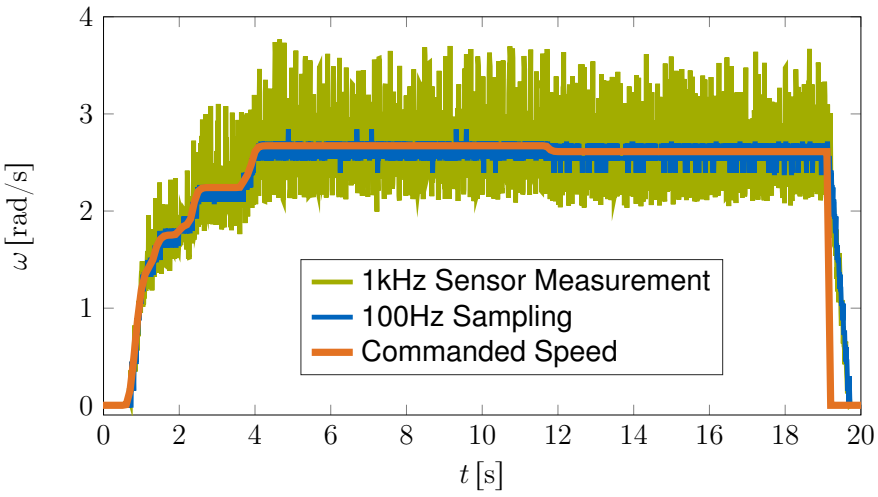

fig. 4.4.: The measured angular wheel velocity of the LRU and the sensor noise for the $1\,\mathrm{kHz}$ and $100\,\mathrm{Hz}$ sampling method.

As mentioned before, the attitude state variables roll α and pitch β cannot be computed by the wheel odometry but have to be taken from other sensors. For wheel odometry, slip estimation and traction control, the roll and pitch are obtained from the output of the fusion EKF, as this data has much less noise than the values from the IMU. If no EKF output is available, the IMU is selected instead.

The sensoric input for the bogie angles ρ_{F} and ρ_{R} is defined as the angle measured at the motors of the SEAs, ignoring the measurements at the spring-damper joints. This design choice was made, as the spring-damper sensor information was not available and the angular deflection is small. Additionally, the velocity $\dot{\rho}_{\mathrm{F/R}}$ is not considered and therefore set to zero, as only slow bogie movements are expected and the velocity measurements would introduce additional noise to the wheel odometry.

Another limitation for the sensors are body oscillations which occur while driving with the LRU and result from the wheel-ground interaction. The oscillations can be seen in the IMU measurements (see figure 4.5) as strong linear accelerations.

In wheel odometry and slip observation, usually a lot of information can be obtained from the torque which is applied by the motors of the wheels. The wheel torque of the LRU is determined by measurements of the motor current and considering the motor constants. Unfortunately, this source of information cannot be easily used for the LRU, as it is strongly influenced by the body oscillations. Even if the kinematic control commands a constant wheel speed, the applied torques are far from constant as the internal motor controller compensates the body oscillations and causes overshoots in the applied torque. This is depicted in figure 4.5 for a single wheel.

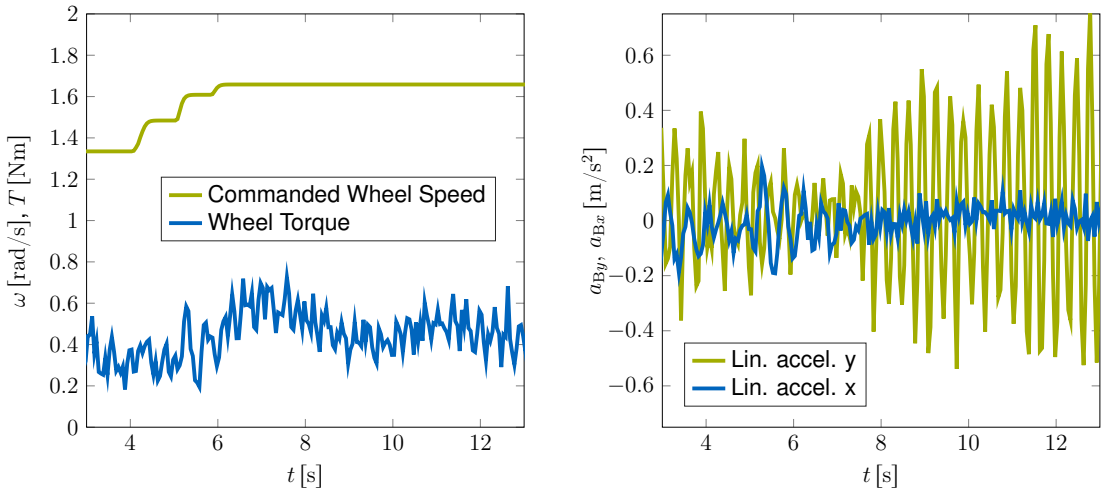

fig. 4.5.: The body oscillations (measured by the IMU, right) strongly influence the torque applied by the wheel motor on a single wheel (left). All signals are filtered for high frequency noise.

Finally, the wheel contact angles ψ_i are set to zero. This is due to the fact, that the LRU does not provide sensors which allow for the measurement of the wheel contact angle, for example sensors like force sensors at the wheel axles. It was attempted to use a procedure proposed by [32] to estimate the wheel contact angles, but it was not yet possible to obtain satisfactory results.

4.4.3. Observation of the Steering Error

Another error in wheel odometry is caused by the elasticity of the rover wheels. In section 2.2 it was outlined that both rovers have wheels with different sets of spokes. The LRU-2 is equipped with 12 steel spokes on each wheel and the LRU-1 with 3 laser sintered titan spokes for each wheel [6].

Both sets of spokes are elastic and experience a deformation while driving in rough terrain. During the process of steering in loose soil, the wheels need to work against the terrain and need to push it aside. This induces side-forces which in return introduce elastic deformations of the wheels in steering direction. As a result, the actual steering angle experiences an offset compared to the commanded steering angle and cannot be measured directly.

This steering offset can be observed however. The torques acting on the wheels at the steering motors are measured and the wheels are modeled as rotational springs with a saturation characteristic for movements around the steering axis.

The characteristic curve of the spring stiffness for LRU-1 is shown in figure 4.6. It was determined by deflecting the wheels by a certain angle and measuring the acting torque. The maximum possible deflection is at $5\,^\circ$, which determined the saturation limit of the curve. The measured points are displayed in the figure together with the determined characteristic curve. The slope was approximated to $2\,^\circ/\mathrm{Nm}$.

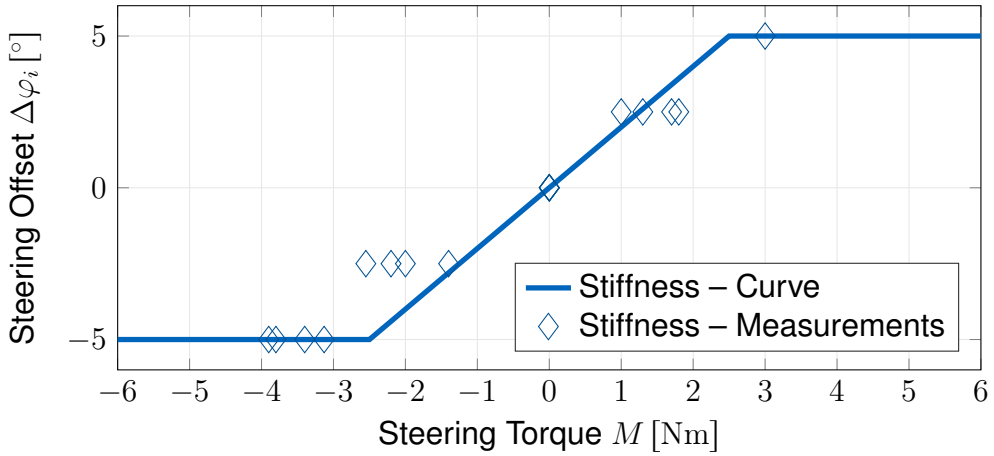

fig. 4.6.: Model of the wheels as torsion spring with saturated stiffness-characteristic – curve and measurements of the rotational stiffness for LRU1.

The 12 spoked wheels of LRU-2 are much more rigid, which renders the influence of the steering error much less significant compared to the LRU-1. Its nonlinear characteristic was therefore set to a slope of $0\,^\circ/\mathrm{Nm}$ and the saturation to $0\,^\circ$, but can be adapted in the future.

4.5. Covariances for Odometry

Before the velocities of the wheel odometry can be integrated into the fusion EKF, the confidence in the measurement needs to be determined and its respective covariance matrix has to be provided to the EKF.

To determine such confidence, the measurement noise of the system inputs and its propagation through the system is considered.

4.5.1. Sensor Covariances

For the wheel odometry, the position measurements for φ_i and $\rho_{\mathrm{F/R}}$ are assumed to be noise free, as these rely on direct encoder measurements with little to no noise. In section 4.4.2, it was determined that the $100\,\mathrm{Hz}$ sampling of the angular wheel velocities ω_i provides results with significantly less measurement noise than the $1\,\mathrm{kHz}$ sampling. However, this noise is still predominant compared to the other inputs and can therefore not be ignored. Evaluating the recorded sensor inputs, the covariance matrix of the measurements turns out to be independent of the velocity and is determined as

$$\boldsymbol{C}_{\omega\omega} = \begin{bmatrix} 0.0076 & 0 & 0 & 0 \\ 0 & 0.0077 & 0 & 0 \\ 0 & 0 & 0.0095 & 0 \\ 0 & 0 & 0 & 0.0097 \end{bmatrix}, \tag{4.43}$$

in $\mathrm{rad}^2/\mathrm{s}^2$. The measurement noise is determined by subtracting the commanded wheel speeds from the measured ones. This noise signal turned out to be zero-mean and its variance is easily computed using MATLAB®. The measurement signals together with the commanded signals can be seen for example in figure 4.4, where the commanded wheel speed (orange) turned out to be equal to the mean of the measured wheel speeds. Note that the sensor signals are derived from encoder readings, and therefore having discrete steps in the amplitude. Thus, the mean is mostly influenced by the distribution of the encoder steps in time, which cannot be seen in 4.4 to to the limited resolution in time-direction. In comparison, the covariance matrix of the $1\,\mathrm{kHz}$ sampling is approximately ten times higher and increases slightly with increasing velocity.

4.5.2. Propagation of Covariances

The forward kinematics in section 4.2 showed, how the body velocity depends on the sensor inputs. A similar dependency between the covariance of the sensor inputs and the body velocity covariance can by shown, following section 3.2.4.

Recall equation (4.35) which describes the relation between the velocities of the center of each wheel and the body velocity. For an individual wheel i, the velocity of the wheel center depends on the angular velocity ω_i and (4.22) can be rewritten to $\boldsymbol{v}_{\mathrm{wc}_i} = \boldsymbol{V}_i\,\omega_i$ with

$$\boldsymbol{V}_i = [r_i\cos(\psi_i), 0, r_i\cos(\psi_i)]^T, \tag{4.44}$$

where ψ_i is the ground contact angle for each wheel i and is not yet considered throughout this work, thus set to zero.

To consider all four wheels, the velocities are stacked together to obtain a matrix $\boldsymbol{V}$, which reads

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{V}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{V}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{V}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{V}_4 \end{bmatrix}. \tag{4.45}$$

Using the matrix $\boldsymbol{V}$ allows to rewrite equation (4.35) to include the angular wheel velocities $\boldsymbol{\omega}$ of all wheels, resulting in

$$\dot{\boldsymbol{\zeta}}_{\mathrm{B}} = \boldsymbol{J}^{+}\boldsymbol{R}_{\mathrm{Bw}}\boldsymbol{V}\boldsymbol{\omega}, \tag{4.46}$$

where $\boldsymbol{R}_{\mathrm{Bw}}$ is the stacked rotation matrix and the assumption $\dot{\rho}_{\mathrm{F/R}} = 0$ is already considered.

Using the previously calculated covariance matrix (4.43) of the angular wheel velocities and following (3.19) allows to determine the covariance matrix of the body velocities as

$$\boldsymbol{C}_{\dot{\zeta}_{\mathrm{B}}\dot{\zeta}_{\mathrm{B}}} = \boldsymbol{J}^{+}\boldsymbol{R}_{\mathrm{Bw}}\boldsymbol{V}\boldsymbol{C}_{\omega\omega}\,\boldsymbol{V}^{T}\boldsymbol{R}_{\mathrm{Bw}}^{T}\boldsymbol{J}^{+T}. \tag{4.47}$$

The covariance matrix of the body velocities obviously depends on the bogie angles $\rho_{\mathrm{F/R}}$ and the steering angles φ_i and is therefore not constant for changing bogie angles and during steering maneuvers.

The computation of the covariances using recorded sensor data showed that the variance for the longitudinal x-velocity varies slightly while driving but generally lays within an order of magnitude of approximately $3 \cdot 10^{-5}\,\mathrm{m}^2/\mathrm{s}^2$. During steering processes, the variance of the y-velocity is correlated to the x-velocity and reaches similar values. The crosscovariance values compute then to approximately $1 \cdot 10^{-6}\,\mathrm{m}^2/\mathrm{s}^2$. For driving without steering, the confidence in y-direction becomes much higher, however no velocity is expected then.

Note, that the covariance of the y-velocity is only relevant while driving in crab mode with a steering angle not equal to zero. In all other cases, especially during ackermann driving, the y-velocity computes to zero anyways.

The LRU currently uses hand-tuned values for the covariance matrix. The hand-tuned values consider the measurements of the body velocity as uncorrelated

with a variance of $1 \cdot 10^{-2}\,\mathrm{m}^2/\mathrm{s}^2$. These values underestimate the confidence in the wheel odometry during laboratory experiments, but are a reasonable assumption for driving experiments in rough terrain as slippage of the wheels is degrading the signal quality significantly. For stand still, the measurements are basically noise free and a higher confidence is assigned to the wheel odometry: $0.00001\,\mathrm{m}^2/\mathrm{s}^2$ for $v_{\mathrm{B},x}$ and $v_{\mathrm{B},y}$.

During this work, the experiments were performed using the hand tuned values. Note that the previously used sensors have a ten times higher sensor noise, and the hand tuned values were designed for such signal input. This might underestimate the measurements of the wheel odometry in the EKF but is still fulfilling the most important criterion: preventing an overconfidence in potentially wrong signals. It can therefore be concluded, that especially under the presence of slip, the hand-tuned values are a reasonable assumption.

Dynamic Covariances: For future applications, the covariances from equation (4.47) can be used. To prevent overconfidence in rough terrain, information on body slip s should be incorporated to scale the covariances accordingly. Resulting for example in

$$\boldsymbol{C}_{\dot{\zeta}_\mathrm{B}\dot{\zeta}_\mathrm{B},\mathrm{slip}} = f(s)\boldsymbol{C}_{\dot{\zeta}_\mathrm{B}\dot{\zeta}_\mathrm{B}}, \tag{4.48}$$

where $f(s)$ is a potentially nonlinear function to model the influence of slip. This however requires careful implementation and thorough testing, as the fusion EKF is sensitive to changes in the provided covariances.

4.6. Experimental Validation

To validate the derived wheel odometry, several tests were performed in the laboratory. Several different trajectories were driven with the LRU-2 and the wheel odometry was computed. To get information about the accuracy of the wheel odometry, the ground-truth information of each trajectory was recorded by tracking the rover using a Vicon® tracking system. Figure 4.7 compares two pairs of trajectories of the LRU-2 before and after the calibration process. The trajectories are displayed in the x-y-ground-plane and refer to the position of the body frame B, which is also called *center of geometry* of the rover.

It can be observed that the position computed by the wheel odometry initially exhibited a significant error, which could be eliminated by calibration.

To show the capabilities of the three-dimensional wheel odometry, a ramp was used as obstacle. A driving maneuver was performed, where the left wheels of the rover drove over the ramp, introducing a combined pitch and roll movement on the rover. The attitude angles α and β were determined from the IMU-measurements. Figure 4.8 shows the ground truth vertical position of the center of geometry and the ground truth pitch angle β compared to the computed ones.

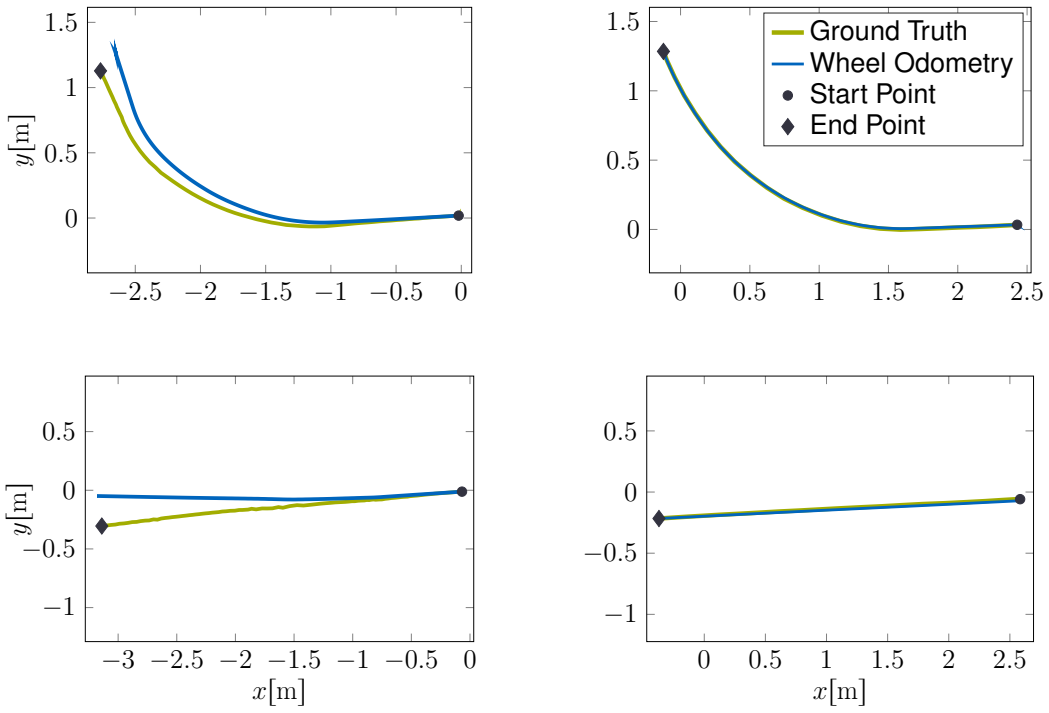

fig. 4.7.: Trajectories driven with the LRU-2 before(left) and after(right) wheel odometry calibration. Comparison of the calculated wheel odometry with the ground truth measurements using the Vicon® tracking system.

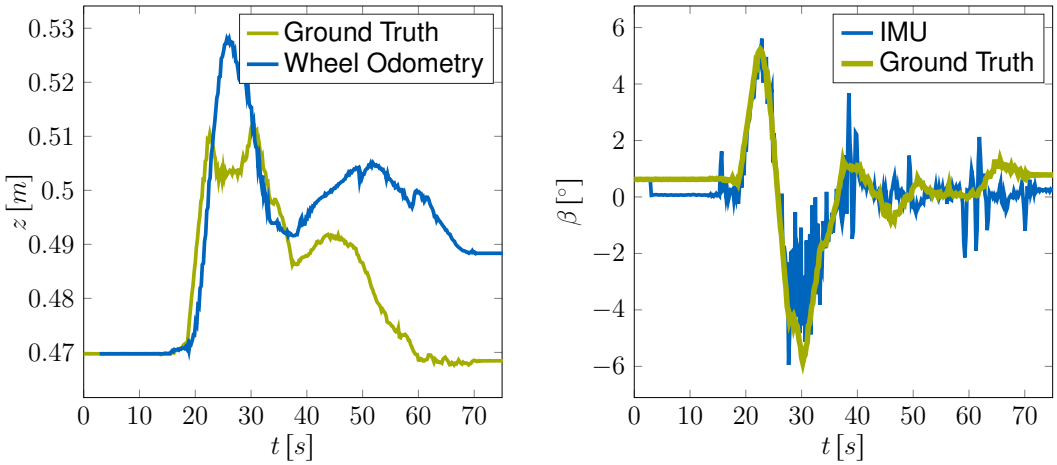

fig. 4.8.: Computed and measured vertical position of the rover while driving over a ramp (left) and the used pitch angle β obtained from the IMU (right).

In the vertical z-direction, an odometry error still occurs because the location of the instantaneous axis of rotation cannot be properly defined, as the wheel contact angles ψ_i are not calculated yet. This could be improved by applying the wheel contact angle estimation proposed by [32]. Additionally, it can be noted that the pitch angles obtained from the IMU match with the ground truth pitch.

5. Slip Estimation

The previous chapter illustrated that one of the major errors in wheel odometry is caused by an imprecise calibration. Even though the calibration of the LRU resulted in an improved, reliable wheel odometry during the laboratory experiments, the situation on loose soil is expected to change drastically.

In chapter 4, the body velocity is calculated under the assumption of an ideal wheel-ground contact. If this assumption does not hold, the wheel velocity will be exposed to a significant error which has to be considered. The deviation of the wheel from the ideal ground contact assumption is called *slip* s. There exist many different definitions for slip, each useful for certain applications. In this work, the definition of slip following [51, p. 14] is used:

$$s = \frac{r\omega - v_{\mathrm{wc}}}{r\omega}, \tag{5.1}$$

where r is the effective rolling radius and ω the angular velocity of a single wheel. The translatory speed of the wheel center is denoted by v_{wc}. This definition provides slip as relative quantity. Alternatively, the difference in velocity can be measured to provide the *slip velocity* $v_{\mathrm{wc,slip}} = r\omega - v_{\mathrm{wc}}$ of a single wheel.

For a vehicle, slip can occur on all wheels simultaneously. The slip of all wheels can be combined and mapped onto the corresponding body frame, allowing to obtain *body slip*. The body slip velocity is defined as [27]

$$\boldsymbol{v}_{\mathrm{B,slip}} = \boldsymbol{v}_{\mathrm{B,WO}} - \boldsymbol{v}_{\mathrm{B,real}}, \tag{5.2a}$$

$$\boldsymbol{\omega}_{\mathrm{B,slip}} = \boldsymbol{\omega}_{\mathrm{B,WO}} - \boldsymbol{\omega}_{\mathrm{B,real}}, \tag{5.2b}$$

or alternatively using the notation with spatial velocities as

$$\dot{\boldsymbol{\zeta}}_{\mathrm{B,slip}} = \dot{\boldsymbol{\zeta}}_{\mathrm{B,WO}} - \dot{\boldsymbol{\zeta}}_{\mathrm{B,real}}, \tag{5.3}$$

and denotes the error between the body velocity computed by the wheel odometry (WO) and the actual velocity.

In rough terrain, while driving over loose soil, slip is expected to occur constantly. This assumption is illustrated in the following by examining the general driving characteristics of wheeled vehicles. The scalar equation of motion in driving direction for a wheeled vehicle is introduced by [51, p. 122] and reads

$$m\ddot{x} = \sum_{i=1}^{4} F_{\mathrm{t},i} - \sum_{i=1}^{4} R_{\mathrm{r},i} - R_{\mathrm{a}} - R_{\mathrm{d}} - R_{\mathrm{g}}, \qquad \text{for} \qquad i = 1,..,4, \tag{5.4}$$

where F_t denotes the tractive effort and R_r the rolling resistance for every wheel individually. On body level, R_a denotes the aerodynamic resistance, R_d the drawbar load, R_g the gravitational pull on the vehicle, m the vehicle mass, and $\ddot{x}$ the acceleration in longitudinal direction. Tractive effort is the result of the motor torque transferred from the wheels to the ground due to shear stress between the wheel surface and the soil. The drawbar load is any force acting on the vehicle due to towing any additional load.

In the case of the LRU, R_a and R_d can be neglected, as the speed of the rover is not high enough for significant aerodynamic effects and no additional load is pulled.

The rolling resistance R_r and the tractive force F_t act on the wheel level and are summed up for the rover body. For a single wheel in loose soil, these forces are illustrated in figure 5.1. The shown wheel rotates with the angular velocity ω and is pressed onto the ground by the gravitational force F_g. This results in friction and allows to apply the motor torque T to the ground via the running surface of the wheel, finally resulting in the tractive force F_t.

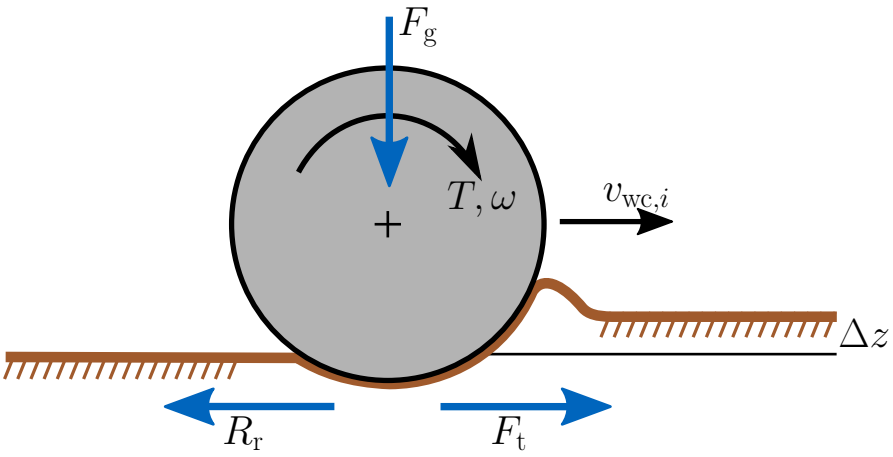

fig. 5.1.: Wheel in loose soil together with the acting forces, adapted from [28].

The forces opposing the tractive force can be summarized as the rolling resistance R_r. In loose soil, the rolling resistance results from vertical and horizontal soil deformations according to [52, p. 450]. The vertical deformations are called sinkage or compaction resistance and appear as the acting gravity pushes the wheel into the loose ground. With additional slip of the wheel, it digs even further into the ground [52, p. 452]. In horizontal direction, the wheel needs to plough through the uppermost layer Δz together with a bow wave of soil that builds up in front of the wheel. The wheel needs to bulldoze the material out of the way and as result an opposing force acts on the wheel, which is therefore referred to as bulldozing resistance [52, p. 452]. Figure 5.1 depicts both the sinkage of the wheels as also the bow wave in direction of the forward velocity $v_{\mathrm{wc},i}$.

Additionally, steep slopes are expected during rough terrain navigation. The gravitational pull R_g acts on the vehicle relative to the steepness, creating fur-

ther forces which act on the wheels in driving direction, adding to resistive forces for uphill drives and opposing these for downhill trajectories. Considering all the forces acting on the wheel and on the vehicle allows to conclude the following impact on the driving dynamics: Except for particular downhill or breaking scenarios, where all resistive forces cancel each other out, a tractive effort F_{t} will always be necessary.

For off-road terrain, [52, p.115,136] introduces two categories of soil, plastic and brittle soils. The shear stress over shear displacement relationship caused by wheels for these two different soil types is shown in 5.2. For both types of soils, it holds that for any desired tractive effort, shear displacement will occur on the wheel level. The distinction between plastic and brittle soil is not necessary for slip estimation, but will be considered later for traction control in chapter 6.

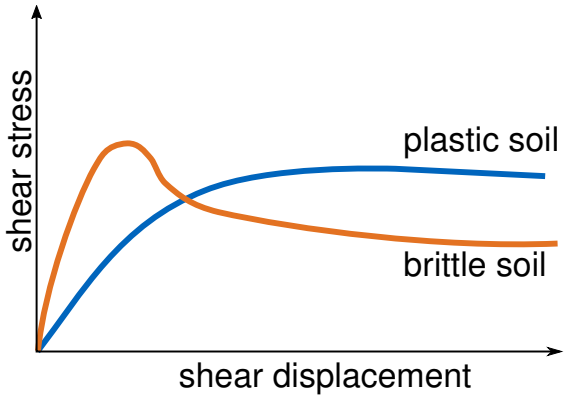

fig. 5.2.: Shear displacement to shear stress characteristic for different types of soil [52, p. 116].

Permanently occurring shear displacement for any commanded tractive effort implies that driving on loose terrain always includes slippage of the wheels. To eliminate the error in pose estimation due to the slip, several concepts of slip measurement and slip estimation are discussed in the following.

5.1. Single Wheel Slip

First of all, single wheel slip is examined. If a drastic change in driving conditions occurs – e.g. a sudden decrease of contact force for a single wheel – this wheel might exhibit a dynamic behavior much different from the other wheels.

There exist several methods to detect such a type of slip. For example, [14] compute the forward kinematic of the wheel odometry to determine the body velocity of the rover (similar to chapter 4). Continuing, they use the inverse kinematics to find the wheel speeds, which are consistent with the body velocity.

Single wheel slip is then detected, if a significant offset between the calculated and measured wheel speeds occurs.

An alternative approach for single wheel slip detection is provided by [18] which relates the measured wheel speed and the measured currents in the wheel hub motors. There, slip is detected, if at constant energy consumption a sudden increase in wheel velocity occurs.

These and other methods all require control of the individual wheels on the motor torque level. However, as outlined in section 2.1, the wheels of the LRU are controlled using kinematic commands. The inverse kinematic algorithm provides commanded wheel speeds to the motor and the internal control of the motor adjusts to the desired speed. If any irregularities occur in the wheel ground contact, the internal motor controllers cancel them out and keep the wheel velocities mostly constant.

As a result of the kinematic control mode, no single wheel slip is expected to occur with the LRU. If situations with very fast dynamics take place, such that the motor controller will not be able to adjust sufficiently fast, single wheel slip is still accounted for by the pseudo inverse in (4.35) [14].

An aspect for future investigation would be to consider the change in energy consumption of a wheel motor while maintaining a constant speed. Decreases in the energy consumption might allow to detect wheel slip.

5.2. Direct Measurement of Body Slip

The previous section treated the case if one single wheel exhibits slip. For pose estimation, it is of much higher importance to determine slip which occurs on all wheels simultaneously.

For odometry correction, it would be favorable to measure body slip directly. The following section introduces two possible methods.

5.2.1. Passive Wheel Slip Detection

It was established at the beginning of this chapter, that slip is mostly caused by the tractive effort of a wheel. Once there is no more torque applied by the wheel motor, the wheel is forced by friction to adjust its speed to the ground truth velocity, which in return can then be measured directly.

The concept is to switch off each wheel in random order for a limited amount of time and power them otherwise. With this approach, a loss in driving performance could be avoided, which would occur by a permanently deactivated wheel. Such a procedure allows to obtain ground truth measurements of the

wheel speeds at certain time steps and is interpolated during the pauses between the measurements.

On the LRU, this concept was tested. While driving, each wheel was switched off for $100\,\mathrm{ms}$ in arbitrary order with $500\,\mathrm{ms}$ periods in between, during which all 4 wheels were fully powered. The speeds of the deactivated wheels could be measured. The trigger signal together with the wheel speed of one activated and one deactivated wheel is depicted in figure 5.3.

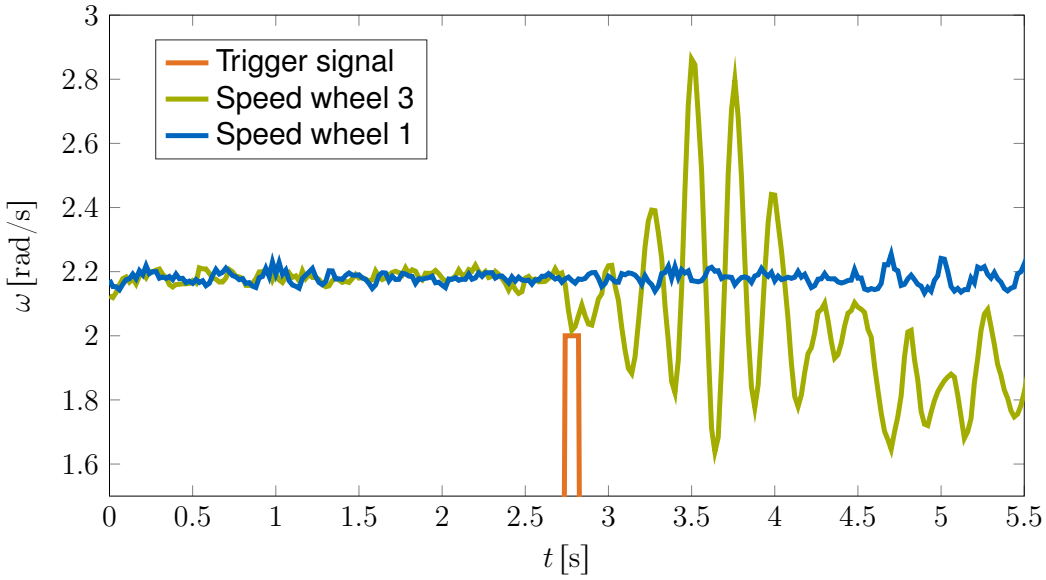

fig. 5.3.: Oscillations at wheel 3 of the LRU-2 as result of the temporary motor deactivation during the period of the trigger signal. The constantly driven wheel 1 is shown for comparison.

Unfortunately, it can be seen that unfavorable oscillations occurred on the wheel level and were additionally transferred onto the rover body. The oscillations are caused by the elastic bogies and wheel suspensions in combination with the motor speed controllers. As long as the wheel motors apply torque to the ground, the resulting forces cause an elastic deformation in the LRU-structure. Setting the motor torque to zero allowed the bogies to snap back to its original position, inducing the oscillations.

Due to the unfavorable oscillations, this approach was dropped for the LRU, but might remain a promising option for more rigid rovers, especially if they are equipped with a higher number of wheels.

5.2.2. Comparison with EKF Output

Another way to directly compute the slip velocity is by comparing the body velocities $\dot{\zeta}_{\mathrm{B,EKF}}$ from the EKF fusion output with the velocities $\dot{\zeta}_{\mathrm{B,WO}}$ of the wheel odometry. The velocities from the sensor fusion are no ground truth information

but are assumed to be much more accurate than the wheel odometry alone. With this approach, the slip velocity is computed as

$$\dot{\boldsymbol{\zeta}}_{\mathrm{B,slip}} = \dot{\boldsymbol{\zeta}}_{\mathrm{B,WO}} - \dot{\boldsymbol{\zeta}}_{\mathrm{B,EKF}}. \tag{5.5}$$

This method is easy to apply and is shown in chapter 6.

However, this method is not suited to be used as input for the fusion filter. If it is used to improve the wheel odometry, it will tune the wheel odometry towards the EKF values instead of the real body velocities, introducing a strong correlation of the signals, potentially degrading the pose estimate. For slippage and traction control however, it is very useful as it allows to obtain good slip estimates for the application of controls.

5.3. Body Slip as Error Model for the Body Velocity

It was established, that body slip is not easily observed directly. However, if it cannot be measured directly, it can be estimated instead.

The real body velocity $\dot{\boldsymbol{\zeta}}_{\mathrm{B,real}}$ can be modeled as the velocity from the wheel odometry combined with an error due to slip. On the body-level of the rover, this can be described as

$$\dot{\boldsymbol{\zeta}}_{\mathrm{B,real}} = \dot{\boldsymbol{\zeta}}_{\mathrm{B,WO}} - \dot{\boldsymbol{\zeta}}_{\mathrm{B,slip}}\left(\boldsymbol{p}, \dot{\boldsymbol{\zeta}}_{\mathrm{B,WO}}, \boldsymbol{f}_{\mathrm{B}}\right), \tag{5.6}$$

where $\dot{\boldsymbol{\zeta}}_{\mathrm{B,WO}}$ is the body velocity computed by the wheel odometry (4.35). The error in the wheel odometry is modeled by $\dot{\boldsymbol{\zeta}}_{\mathrm{B,slip}}$, where the slip-error-model depends on the rover velocity, the acting body forces $\boldsymbol{f}_{\mathrm{B}}$, and is additionally parameterized by the parameter-set $\boldsymbol{p}$. Recall from (4.20) that the body velocity consists of linear and angular components, denoted by $\boldsymbol{v}_{\mathrm{B}}$ and $\boldsymbol{\omega}_{\mathrm{B}}$, respectively.

In [27] and [21], such parameterized velocity-error-models for the body slip velocity are introduced together with a method to determine the used set $\boldsymbol{p}$ of parameters. There, the parameters weight the influence of different phenomena like steep slopes or rolling resistance in the error-model and are determined according to the respective soil characteristics.

For this work, a parameterization following [21] is chosen and the slip velocity on body level is modeled as

$$\boldsymbol{v}_{\hat{\mathrm{B}},\mathrm{slip}} = \left(p_1 \frac{f_{\hat{\mathrm{B}},x}}{f_{\hat{\mathrm{B}},z}} v_{\hat{\mathrm{B}},x} + p_2 v_{\hat{\mathrm{B}},x}\right) \boldsymbol{e}_{\hat{\mathrm{B}},x} + p_3 \frac{f_{\hat{\mathrm{B}},y}}{f_{\hat{\mathrm{B}},z}} v_{\hat{\mathrm{B}},x} \boldsymbol{e}_{\hat{\mathrm{B}},y}, \tag{5.7}$$

and

$$\boldsymbol{\omega}_{\mathrm{B,slip}} = \left(p_4 \frac{f_{\hat{\mathrm{B}},y}}{f_{\hat{\mathrm{B}},z}} v_{\hat{\mathrm{B}},x} + p_5 v_{\hat{\mathrm{B}},x} + p_6 \omega_{\mathrm{B},z}\right) \boldsymbol{e}_{\mathrm{B},z}. \tag{5.8}$$

The parameters p_1 and p_3 model linear slip, which results from driving on steep slopes, p_2 models the rolling resistance of the soil, p_4 and p_6 account for over-steering/understeering and p_5 models any asymmetry between the left and right side of the rover [21].

Additionally, $\boldsymbol{f}_{\hat{\mathrm{B}}}$ and $\boldsymbol{e}_{\hat{\mathrm{B}}}$ stand for the body-forces and the unit vectors respectively. The input velocities are the velocities $\boldsymbol{v}_{\hat{\mathrm{B}}}$ and $\boldsymbol{\omega}_{\hat{\mathrm{B}}}$ from the wheel odometry. The slip velocity is expressed in the newly introduced *body-drive-frame* $\hat{\mathrm{B}}$. In this frame, the $\hat{z}$-axis of the drive-frame is identical with the z-axis of the original body frame, but the $\hat{x}$-axis points in direction of the momentary body-velocity.

Describing the slip velocity and the forces in the body-drive-frame is necessary to account for lateral linear velocities which can result from one of the two different driving modes of the LRU: the crab-mode (see section 2.1). In the body drive frame, the velocity of the rover measured by the wheel odometry reads zero in $\hat{y}$ direction. This frame is used to ensure, that the influence of the parameters, for example rolling resistance, is always computed with respect to the driving direction of the rover and therefore agrees with the orientation of the wheels.

The transformation of the velocities and forces into the drive frame is computed via the rotation matrix $\boldsymbol{R}_{\hat{\mathrm{B}}\mathrm{B}}(\sigma)$ around the z-axis with the rotation angle σ defined as

$$\sigma = \mathrm{atan2}(v_{\mathrm{B},y}, v_{\mathrm{B},x}). \tag{5.9}$$

Additionally, the resulting linear slip velocity can be transformed back into the original body frame by

$$\boldsymbol{v}_{\mathrm{B,slip}} = \boldsymbol{R}^T_{\hat{\mathrm{B}}\mathrm{B}}(\sigma)\boldsymbol{v}_{\hat{\mathrm{B}},\mathrm{slip}}. \tag{5.10}$$

For angular velocities, the error model only considers the component around the z-axis which therefore requires no transformation.

The body forces can either be obtained by using the linear acceleration from the IMU as

$$\boldsymbol{f}_{\mathrm{B}} = \frac{1}{2}\left(\boldsymbol{R}^T_{\mathrm{BF}} + \boldsymbol{R}^T_{\mathrm{BR}}\right)\boldsymbol{M}\boldsymbol{a}_{\mathrm{IMU}}, \tag{5.11}$$

where the accelerations $\boldsymbol{a}_{\mathrm{IMU}}$ are measured by the IMU and combine all inertial, gravitational, and centrifugal terms and $\boldsymbol{M} = \boldsymbol{I}m$ denotes the rover mass.

Alternatively, assuming quasi-static dynamics of the rover, the body forces rely on gravitation and Coriolis terms only and can be estimated following [53], using

$$\boldsymbol{f}_{\mathrm{B}} = \frac{1}{2}\left(\boldsymbol{R}^T_{\mathrm{BF}} + \boldsymbol{R}^T_{\mathrm{BR}}\right)\left(\boldsymbol{M}\boldsymbol{g}_{\mathrm{B}} - \boldsymbol{\omega}_{\mathrm{B}} \times \boldsymbol{M}\boldsymbol{v}_{\mathrm{B}}\right), \tag{5.12}$$

which yields a signal with much less measurement noise compared to the IMU signal and is therefore used for the slip estimation. Here, $\boldsymbol{g}_{\mathrm{B}}$ is the gravity vector expressed in the body frame and obtained from the attitude that is provided by the EKF. For $\boldsymbol{v}_{\mathrm{B}}$ and $\boldsymbol{\omega}_{\mathrm{B}}$, the velocities from the wheel odometry are used as they are already available and the resulting error is assumed to be negligible [21]. The rotation $\boldsymbol{R}^T_{\mathrm{BF/BR}}$ between the body and the bogies is additionally considered to provide the correct lateral forces that act in the driving plane. Note

that this is a simplified computation which holds for the current rover configuration, as both bogie angles are kept at $\rho_{\mathrm{F}} \approx \rho_{\mathrm{R}}$ and the rotations are averaged to account for measurement inaccuracies.

5.3.1. Obtaining the Slip Parameter

To provide correct information on the slip velocities, an accurate set of parameters $\boldsymbol{p}$ is required. One way to determine the set of parameters is the Integrated Prediction Error Minimization (IPEM) approach presented in [27].

The idea of IPEM is to take a general model of a (possible nonlinear) dynamic system and enhance it with an additional set of parameters $\boldsymbol{p}$, which are used to account for modeling errors:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{p})), \tag{5.13a}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}). \tag{5.13b}$$

Using system identification techniques, these parameters can be determined and the modeling errors can be minimized. Note, that only the input vector $\boldsymbol{u}$ is parameterized, an assumption made by [27] to simplify the modeling. To account for parameters that act directly on the system and not on the inputs, the input vector can be augmented with these parameters as additional, constant inputs [27].

IPEM allows to identify the parameters by comparing the estimated (predicted) output of the system model with the measured output of the real system. The main advantage of IPEM compared to other parameter optimization methods is that in experiments for the parameter identification, no continuous information on the real trajectory is necessary. It is only required to measure the real start and end point and to record all of the system inputs $\boldsymbol{u}$ [27].

To focus on the start and end points, the integrated dynamics over a whole trajectory are considered. These can be written as

$$\begin{aligned} \boldsymbol{x}(t_{\mathrm{end}}) &= \boldsymbol{x}(t_0) + \int_{t_0}^{t_{\mathrm{end}}} \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{u}(\boldsymbol{p}, \tau))\mathrm{d}\tau \\ &= \boldsymbol{g}(\boldsymbol{x}(t_0), \boldsymbol{u}(\boldsymbol{p}, ..)), \end{aligned} \tag{5.14}$$

where $\boldsymbol{u}(\boldsymbol{p}, ..)$ denotes the system input for all $\tau \in [t_0, t_{\mathrm{end}}]$.

The parameters $\boldsymbol{p}$ are determined by a parameter identification experiment, which requires a series of trajectories to be performed. The start and end points of each trajectory are measured. Additionally, the system model is used to determine the estimated end point of the trajectory by considering the recorded system inputs $\boldsymbol{u}$ and the starting point $\boldsymbol{x}(t_0)$.

The measurements provide the *ground truth* information on the trajectory end points and are denoted as $\boldsymbol{y}_{\mathrm{meas}}$. The observation of the estimated end point

depends on the parameters $\boldsymbol{p}$ and is written as $\boldsymbol{h}(\boldsymbol{p})$. Considering the error $\tilde{\boldsymbol{y}}$ between the estimated and the real end point observations, following relation holds [27]:

$$\tilde{\boldsymbol{y}} = \boldsymbol{y}_{\text{meas}} - \boldsymbol{h}(\boldsymbol{p}) = \boldsymbol{H}_{\text{sys}} \Delta \boldsymbol{p}, \tag{5.15}$$

where $\Delta \boldsymbol{p}$ is the required change in parameters to minimize the estimation error $\tilde{\boldsymbol{y}}$. The matrix $\boldsymbol{H}_{\text{sys}}$ is called the *parameter Jacobian matrix* of a trajectory and models the influence that each parameter has on the estimated end point of the corresponding trajectory.

The influence of the parameters on the estimated end of the trajectory is usually nonlinear and the parameter Jacobian matrix is the linearized representation, with [27]

$$\boldsymbol{H}_{\text{sys}} = \left[\frac{\mathrm{d}\boldsymbol{g}}{\mathrm{d}p_1}, ..., \frac{\mathrm{d}\boldsymbol{g}}{\mathrm{d}p_n} \right]. \tag{5.16}$$

The crucial task for IPEM is to determine the parameter Jacobian matrix, which is later explained in detail.

In the case of slippage estimation for the LRU, $\boldsymbol{x}(t_0)$ and $\boldsymbol{x}(t_{\text{end}})$ represent the start and end position and orientation of the rover body during a driving experiment, expressed in the inertial frame. The system input $\boldsymbol{u}$ is the velocity output (5.6) of the wheel odometry in the body frame after applying the slip-error-model and is recorded as $\boldsymbol{u}(\boldsymbol{p}, ..)$ as long as the LRU drives from the start to the end point of the trajectory.

The nonlinear function $\boldsymbol{f}$ contains the body velocity inputs and the transformation into the inertial frame. The general transformation reads

$$\boldsymbol{T}_{\text{IB}} = \begin{bmatrix} \boldsymbol{R}_{\text{IB}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{W}_{\text{IB}} \end{bmatrix} \tag{5.17}$$

and allows to transform both linear and angular velocities into the inertial frame. The velocities expressed in the inertial frame are then integrated to obtain the pose.

The pose of the LRU at the end point of each trajectory can be directly observed, which yields

$$\boldsymbol{h}(\boldsymbol{p}) = \boldsymbol{x}_{\text{pred}}(t_{\text{end}}) = \boldsymbol{x}_{\text{WO+slip}}(t_{\text{end}}) \tag{5.18}$$

for the estimated system output and

$$\boldsymbol{y}_{\text{sys}} = \boldsymbol{x}_{\text{meas}}(t_{\text{end}}) = \boldsymbol{x}_{\text{GT}}(t_{\text{end}}) \tag{5.19}$$

for the measured ground truth (GT) end point.

Recall that it is the task to determine a set of parameters $\boldsymbol{p}$ in such way that the used error-model allows to minimize the error between the measured and the estimated end point of a trajectory. Usually, the number of parameters is higher than the number of DOF determined by an individual end point measurement. To find a suited set of parameters, it is therefore necessary to consider several

trajectories with their respective end point measurements [27]. Indeed, the more trajectories are available, the more generalized parameters are found, which avoids overfitting.

Having the measurements of several trajectories available, renders equation (5.15) for $\Delta \boldsymbol{p}$ overdetermined. As it was outlined in chapter 3, recursive and batch techniques are available to find $\Delta \boldsymbol{p}$ in an overdetermined system. A recursive, online way of determining $\Delta \boldsymbol{p}$ while driving with the rover is briefly discussed in section 5.3.5.

To determine the parameters offline, a set of N driven trajectories is recorded. For all trajectories, the parameter Jacobian matrices are stacked, and equation (5.15) results an overconstrained system

$$\Delta \boldsymbol{p} = \begin{bmatrix} \boldsymbol{H}_{\text{sys}}^1 \\ . \\ . \\ . \\ \boldsymbol{H}_{\text{sys}}^N \end{bmatrix}^{\#} \begin{bmatrix} \boldsymbol{y}_{\text{sys}}^1 - \boldsymbol{h}(\boldsymbol{p})^1 \\ . \\ . \\ . \\ \boldsymbol{y}_{\text{sys}}^N - \boldsymbol{h}(\boldsymbol{p})^N \end{bmatrix}, \tag{5.20}$$

which is solved using the generalized inverse [27]. If the ground truth measurements are obtained using different methods or with different accuracy, each trajectory can be weighted accordingly.

The parameter set $\boldsymbol{p}$ is then determined iteratively over several update steps k with

$$\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + \nu \, \Delta \boldsymbol{p}, \tag{5.21}$$

where $\nu \in [0, 1]$ denotes the update rate.

5.3.2. Parameter Jacobian Matrix

For the use of IPEM, obtaining the parameter Jacobian matrix is the crucial task. According to [27], the most straightforward way to obtain this matrix is done by computing the derivative numerically. There,

$$\frac{\mathrm{d}\boldsymbol{g}}{\mathrm{d}p_i} \approx \frac{\boldsymbol{g}(\boldsymbol{x}(t_0), \boldsymbol{u}(\boldsymbol{p} + \delta \boldsymbol{p}_i, ..)) - \boldsymbol{g}(\boldsymbol{x}(t_0), \boldsymbol{u}(\boldsymbol{p}, ..))}{\epsilon} \tag{5.22}$$

is the derivative of $\boldsymbol{g}$ with respect to a parameter p_i and is computed by applying the forward difference with a small ϵ. For each i, the vector $\delta \boldsymbol{p}_i$ introduces a small change ϵ on the ith parameter, keeping all other parameters constant. This approach is easy to implement but has as disadvantage that it is computationally expensive, for the reason that the trajectory needs to be computed repeatedly, once for each variation $\delta \boldsymbol{p}_i$.

A computationally less expensive form of finding the parameter Jacobian is presented by [27] and is based on the work of [26]. There, the propagation of the

odometry error through the system is modeled and linearized. Using a convolution integral, the influence of errors can be calculated over the length of the trajectory, which allows to determine the parameter Jacobian matrix.

This alternative approach is derived at first and later adapted to the LRU. First of all, the system (5.13) is linearized, resulting in the Jacobian matrices $\boldsymbol{F}$ and $\boldsymbol{G}$:

$$\boldsymbol{F} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}, \quad \boldsymbol{G} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}. \tag{5.23}$$

Following the theory of linear systems, such systems have the general solution (3.4) with the state transition matrix $\boldsymbol{\Phi}$. Determining the state transition matrix usually requires integration of the matrix exponential (3.5). However, for any matrix $\boldsymbol{F}$ with no entries on the diagonal and below (strict upper rectangularity), [26] shows that the computation (3.5) of the state transition matrix can be simplified to an integral with

$$\boldsymbol{\Phi}(t, \tau) = \boldsymbol{I} + \int_{\tau}^{t} \boldsymbol{F}(\hat{\tau}) \mathrm{d}\hat{\tau}. \tag{5.24}$$

To allow for a fast computation of the parameter Jacobian matrix, it is therefore beneficial to find a system model which allows for a strict upper rectangularity in $\boldsymbol{F}$. Such model can be found for the LRU, which is shown later on.

Using the linearized system matrices (5.23) and the general solution (3.4), allows to rewrite (5.14) as

$$\begin{aligned} \boldsymbol{x}(t_{\mathrm{end}}) &= \boldsymbol{g}(\boldsymbol{x}(t_0), \boldsymbol{u}(\boldsymbol{p}, ..)) \\ &= \boldsymbol{\Phi}(t_{\mathrm{end}}, t_0)\boldsymbol{x}(t_0) + \int_{t_0}^{t_{\mathrm{end}}} \boldsymbol{\Phi}(t_{\mathrm{end}}, \tau)\boldsymbol{G}(\tau)\boldsymbol{u}(\boldsymbol{p}, \tau)\mathrm{d}\tau. \end{aligned} \tag{5.25}$$

This is inserted into (5.16), the derivative is moved inside the integral. The resulting expression for the parameter Jacobian matrix can therefore be written as [27]

$$\boldsymbol{H}_{\mathrm{sys}} \approx \int_{t_0}^{t} \boldsymbol{\Phi}(t, \tau)\boldsymbol{G}(\tau) \frac{\mathrm{d}\boldsymbol{u}(\boldsymbol{p}, \tau)}{\mathrm{d}\boldsymbol{p}} \mathrm{d}\tau. \tag{5.26}$$

Note, that this is an approximation of the parameter Jacobian matrix for nonlinear systems, as the linearized system matrices (5.23) are used.

It was established that the system input $\boldsymbol{u}$ resembles the real rover velocity $\dot{\boldsymbol{\zeta}}_{\mathrm{B,real}}$ on the body level. Equation (5.6) is rewritten to express $\boldsymbol{u}$ as

$$\boldsymbol{u}(\boldsymbol{p}, t) = \dot{\boldsymbol{\zeta}}_{\mathrm{WO}}(t) - \boldsymbol{C}(t)\boldsymbol{p}, \tag{5.27}$$

where the slip velocity is displayed as product of the *parameter influence matrix* $\boldsymbol{C}$ and the parameter vector $\boldsymbol{p}$. This subsequently yields the parameter Jacobian matrix as

$$\boldsymbol{H}_{\mathrm{sys}} \approx -\int_{t_0}^{t} \boldsymbol{\Phi}(t, \tau)\boldsymbol{G}(\tau)\boldsymbol{C}(\tau)\mathrm{d}\tau. \tag{5.28}$$

The parameter influence matrix depends on the chosen slip parameterization and is explained in detail in the following.

Using (5.28) reduces the number of required integrations significantly. Instead of one integration per parameter, only the convolution integral needs to be computed. This is especially of importance for sets with high numbers of parameters.

5.3.3. System Model for Parameter Estimation

The previous section provided the general method to determine the parameter Jacobian matrix $\boldsymbol{H}_{\mathrm{sys}}$. This is now applied to the LRU, specifically considering the presented slip parameterization (5.7) and (5.8).

For this work, a simplification is used, which finds the slip parameters by using a two-dimensional framework to compute the trajectories and measure the start and end points. This requires the use of the 3 DOF pose $\boldsymbol{\xi}$ instead of the 6 DOF pose $\boldsymbol{\zeta}$.

In this work, the parameter influence matrix

$$\boldsymbol{C} = \begin{bmatrix} \frac{f_{\hat{\mathrm{B}},x}}{f_{\hat{\mathrm{B}},z}} v_{\hat{\mathrm{B}},x} & v_{\hat{\mathrm{B}},x} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{f_{\hat{\mathrm{B}},y}}{f_{\hat{\mathrm{B}},z}} v_{\hat{\mathrm{B}},x} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{f_{\hat{\mathrm{B}},y}}{f_{\hat{\mathrm{B}},z}} v_{\hat{\mathrm{B}},x} & v_{\hat{\mathrm{B}},x} & \omega_{\hat{\mathrm{B}},z} \end{bmatrix} \tag{5.29}$$

is used, which got derived from the model of the slip velocity (5.7) and (5.8) and considers the two-dimensional simplification. The system input is determined as

$$\boldsymbol{u}(\boldsymbol{p}, t) = \begin{bmatrix} v_{\mathrm{B},x,\mathrm{WO}}(t) \\ v_{\mathrm{B},y,\mathrm{WO}}(t) \\ \omega_{\mathrm{B},z,\mathrm{WO}}(t) \end{bmatrix} - \boldsymbol{C}(t)\boldsymbol{p}, \tag{5.30}$$

by considering the two-dimensional case of (5.27), with the selected components from the 6 DOF wheel odometry. The dynamic system (5.13a) then reads

$$\dot{\boldsymbol{\xi}} = \boldsymbol{T}_{\mathrm{IB}} \boldsymbol{u}(\boldsymbol{p}, t), \tag{5.31}$$

where $\boldsymbol{T}_{\mathrm{IB}}$ is the transformation matrix in (4.41) that transforms the velocities from the body to the inertial frame.

It can be seen that the input coupling matrix $\boldsymbol{G}$ is identical to $\boldsymbol{T}_{\mathrm{IB}}$ and reads

$$\boldsymbol{G} = \boldsymbol{T}_{\mathrm{IB}} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.32}$$

The dynamic matrix $\boldsymbol{F}$ is found by following (5.23) and is determined as

$$\boldsymbol{F} = \begin{bmatrix} 0 & 0 & -u_x \sin(\theta) - u_y \cos(\theta) \\ 0 & 0 & u_x \cos(\theta) - u_y \sin(\theta) \\ 0 & 0 & 0 \end{bmatrix}, \tag{5.33}$$

where u_x and u_y are the linear velocity components of the input $\boldsymbol{u}$. It can be seen, that $\boldsymbol{F}$ fulfills the condition for strict upper rectangularity, which allows to determine the transition matrix according to (5.24). The resulting transition matrix models the influence of the wheel odometry velocities on the resulting pose estimate during a time interval $[\tau, t]$ and can be written as [27]

$$\boldsymbol{\Phi}(t, \tau) = \begin{bmatrix} 1 & 0 & -(y(t) - y(\tau)) \\ 0 & 1 & x(t) - x(\tau) \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.34}$$

The meaning of this matrix can be illustrated by an example: An error in the measurement of the angular velocity at the beginning of the trajectory causes an higher error in the final position as an angular error occurring close to the end. This different influence is considered by the difference of the actual to the final position in the last row of the transition matrix.

It can be seen that in (5.7) and (5.8) only the x, y and yaw velocity of the rover body are considered as input $\boldsymbol{u}$. For instantaneous movements, the rover is mostly limited to a plane tangential to the ground surface, which renders estimation of slip in other directions insignificant [21]. This assumption can be applied to the LRU, as the pitch and roll movements are small compared to the yaw movement and due to the small bogie angles ρ a velocity in z direction is much smaller than in y and the slip in this direction is chosen to be neglected.

For IPEM, the body velocities first need to be transformed to the inertial frame using the input coupling matrix $\boldsymbol{G}$ and are then integrated to determine the pose. Recall, that for the application on the LRU, a two-dimensional, 3 DOF pose $\boldsymbol{\xi}$ is considered.

Using such a framework provides several advantages, first of all that an analytical solution for $\boldsymbol{\Phi}$ can be found, which reduces the computational costs significantly. The simplification to use a two-dimensional position allows for a broader range of ground truth measurement methods to be used, as the vertical component can be ignored. Errors, which occur due to unavailable information on the vertical component can be minimized by performing the driving experiments on a plane area with a significant slope angle. Aligning the inertial frame with the plane allows for precise two-dimensional measurements while still considering the influence of the slope on slip - which can be determined by driving both in parallel to the highest-slope gradient as well as perpendicular to it.

If ground-truth measurements are available for both the horizontal and also vertical components of the position, the system model for determining the parameter could be augmented.

Alternative System Models: A first step is an improved two-dimensional model proposed by [21], where the transformation $\boldsymbol{T}_{\mathrm{IB}}$ of the velocities into the inertial frame also considers pitch and roll of the rover. For this, the current, two-dimensional plane of movement of the rover gets mapped onto the three-dimensional terrain by considering the roll and pitch, while still limiting the ground truth measurement to the x, y, and yaw values. The transformation $\boldsymbol{T}_{\mathrm{IB}}$ is hereby partly the 6 DOF transformation from (4.39), considering only the rows for the x, y, and yaw values and the columns for the 3 DOF body velocities. This allows to keep the state transition matrix (5.34) while already considering the full three-dimensional character of the terrain. The advantage of this method is that it avoids a too strong z-error, if the trajectory ends in areas where the surface does not longer align with the x-y-plane of the inertial frame. Note that still only the x and y values of the measurement are considered in the parameter estimation, but it is necessary to determine the three-dimensional position exactly to distinguish between z and y.

Another possible system model for the parameter estimation would be the use of the full 6 DOF model. This would require full three-dimensional ground truth measurements and would additionally complicate the determination of the parameter Jacobian matrix, as the dynamic matrix $\boldsymbol{F}$ for a 6 DOF model resulting from the transformation matrix in (4.39) would clearly not fulfill the criterion for strict upper triangularity. Thus, the state transition matrix needs to be found by either computing the matrix exponential in the convolution integral or by relying on the numerical computation (5.22).

During the Etna 2017 parameter calibration experiments for parameter estimation, no means for three-dimensional ground truth measurement were available and a fast computation of the parameters was desired. Therefore, the 3 DOF slip parameterization was chosen. However, this could be adjusted for future experiments.

Alternative Parameter Sets: Additional parameter sets to model the slip velocity have also been tested. Sets of the mentioned six parameters together with a seventh parameter, once to account for acceleration and once to distinguish between upwards and downwards slopes did not improve the error estimate.

A parameter set of 13 parameters used by [27] provided only little improvement to the described 6 parameters, and can be attributed to possible overfitting. This parameter set is very similar to the six parameters used in this thesis. The main difference is, that it considers Coriolis forces and gravitational forces individually instead of the overall body forces. In such case it is always of advantage to select the smaller parameter set, as it requires less recorded trajectories for calibration.

So far, the slip model (5.7) and (5.8) only considered linear velocities. Future work on the slip parameterization could include investigation of possible nonlinear influences on the slip, as observations while driving on-site suggested such correlation.

5.3.4. Experiment: Parameter Estimation

During the ROBEX moon analogue mission, the slip parameter estimation was performed on LRU-1 for the soil on Mt. Etna. For the batch parameter estimation, a set of 17 trajectories were recorded. While driving, the sensory input was logged and the start and end points were measured manually. By that time, no other ground truth measurement method was available than a measurement tape, which restrained the parameter estimation to the 3 DOF case.

To allow for a precise determination of the rover pose, redundant information was recorded. A coordinate system was constructed at a moderately steep flank of the mountain with a slope between 5° to 9° and little change in slope around the origin. The axes of the coordinate system were built by spanning a right triangle using constructive cord, which got fixed to the ground by tent pegs. The right angle was verified by measuring the hypotenuse of the triangle and comparing these to the catheti following Pythagoras.

From the base line of the coordinate system, the distance to each of the rover wheel centers were measured by the measurement tape. Knowing the rover geometry, the vector between two wheels each was calculated and from the mid a perpendicular was dropped to determine the rover center. The yaw angle is calculated comparing the dropped perpendicular to the axis of the coordinate system. The measured position of all four wheels together provided redundant information on the rover pose, which in turn was used to minimize the error using a least squares approach for the several calculated center points.

Figure 5.4 shows LRU-1 positioned close to the origin of the constructed inertial frame. The green constructive cord which defines the frame-axis is pictured as well.

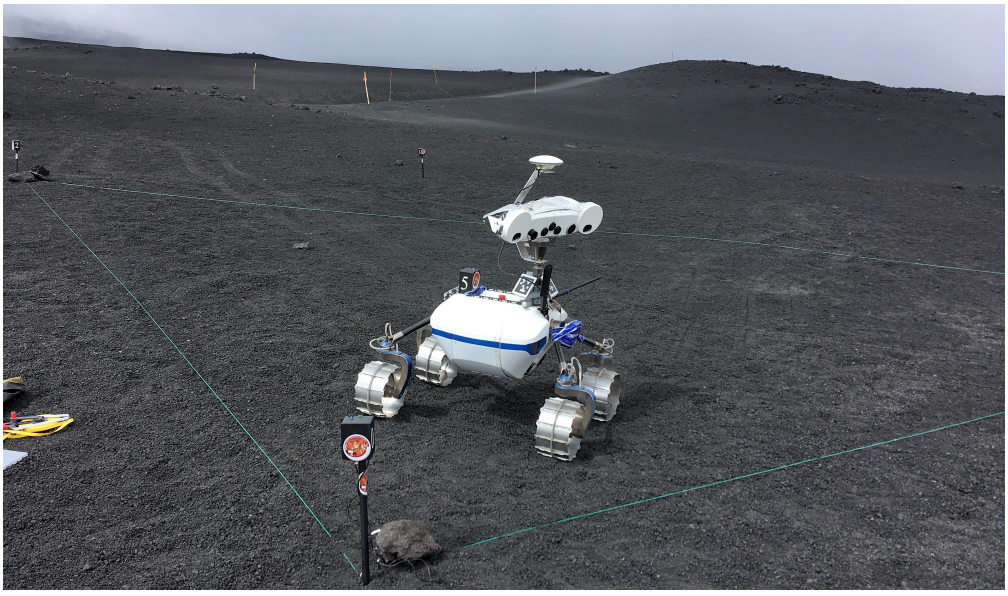

fig. 5.4.: The LRU-1 standing close to the origin of the constructed coordinate frame for the slip-parameter-calibration experiment on Mt. Etna.

Due to the construction of the frame at the inclined mountain flank, the influence of the slope on slip can be conveniently defined by performing driving maneuver in direction of the slope gradient and perpendicular to it. Note that the inertial frame is defined parallel to the hill slope to reduce the error resulting from the two-dimensional assumption.

The 17 recorded trajectories are displayed in figures 5.5 and 5.6. Orange is the trajectory computed by the wheel odometry without considering the slip error. Blue denotes the trajectory of the wheel odometry with the parameterized error model, where the utilized six parameters are the determined by IPEM and are listed in (5.35). The ground truth start and end points are denoted as black circle or diamond respectively. Finally, the trajectory computed by the EKF filter output is provided as well. As the filter output is to be assumed much more precise, it is a good reference for the other trajectories, even though this filter output is computed using the wheel odometry without slip-compensation.

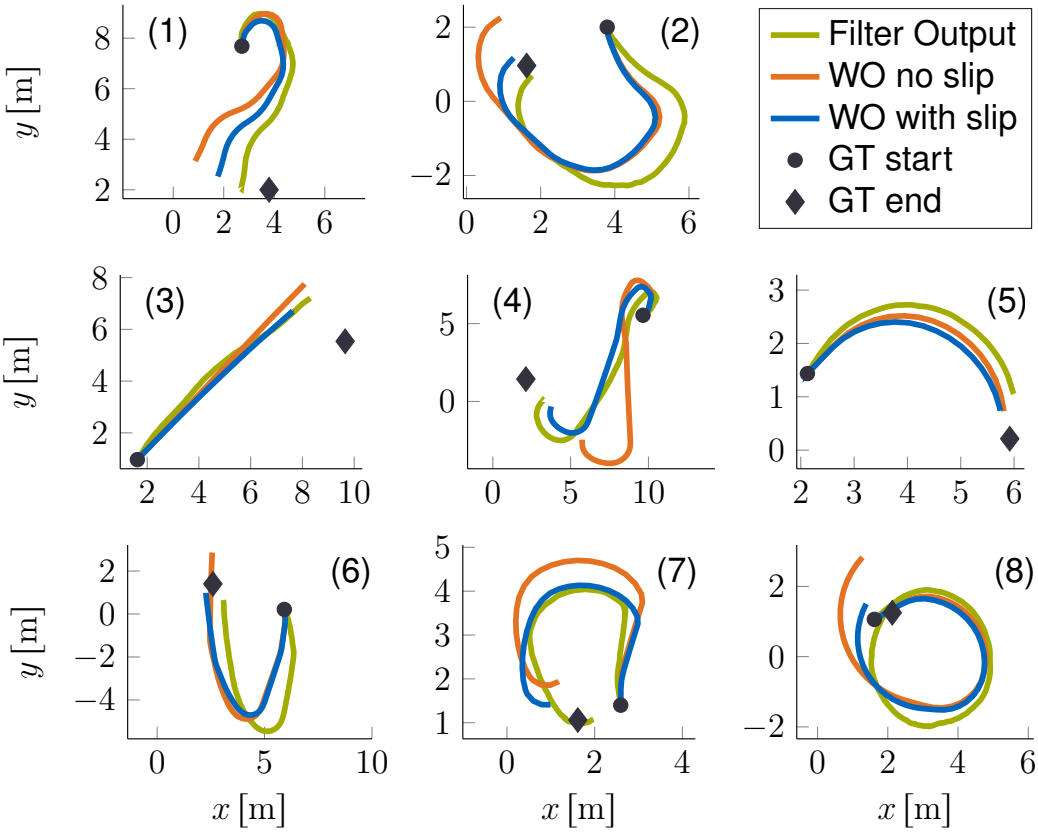

fig. 5.5.: First set of trajectories used for the calibration of slip parameters. Ground truth (GT) start and end points, filter output, and wheel odometry (WO) are shown.

Note that one error is not modeled by this method: The experiments (2) and (6) show an initial yaw offset between wheel odometry and the filter output, which is probably caused by steering during standstill and cannot be observed by the wheel odometry: to steer, the wheels have to move loose soil sideways, which can introduce a rotation of the rover body. Comparing plots (11) and (12)

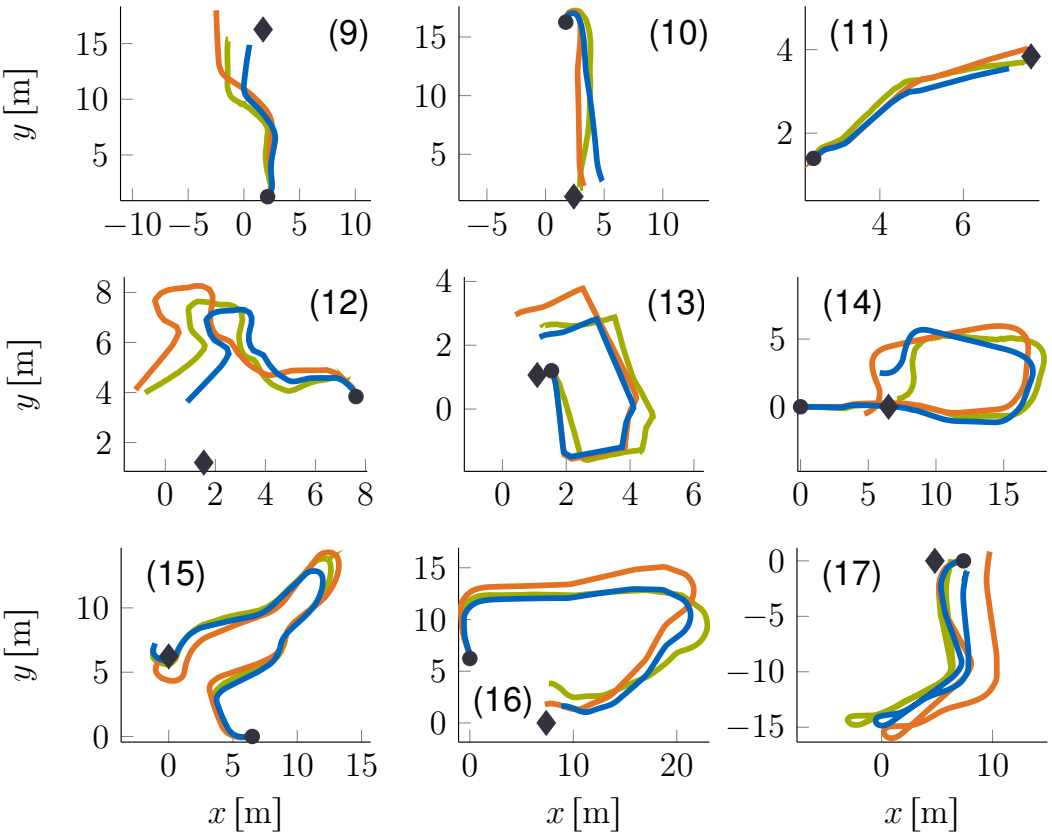

fig. 5.6.: Second set of trajectories used for the calibration of slip parameters. Ground truth (GT) start and end points, filter output, and wheel odometry (WO) are shown.

also shows, that the assumption on homogeneous soil does not hold globally as in plot (12) there is significantly more slip in global y direction than in plot (11). During the experiments (3), (12) and (13), the filter output performs quite poorly, which can be contributed to a wrong orientation of the camera system and disturbances by persons passing in front of the cameras, which in turn forces the filter to rely more on the wheel odometry.

However, it can be seen in general that the slip parameterization in the velocity-error-model allows for better position estimation than without the slip model.

Note that the last four trajectories are significantly longer than the others. For these, the body center position was measured directly and the orientation was determined by driving the rover over the axis of the coordinate system and aligning it in parallel to the axis – a method less accurate than the previous one. It is assumed, that for parameter estimation, the shorter trajectories are more important, as each was performed in such way to model specific influences on slip. To account for this importance and for the higher accuracy in measurement, the first 13 trajectories are given a weight of 1 and the others a weight of 0.5 in the weighted least squares batch estimation (5.20).

The batch calibration is performed over several iterations following (5.21) with an update rate of $\nu = 0.7$

for $k = \{0, ..., 10\}$ iterations, with $\boldsymbol{p}_0 = \mathbf{0}$, where convergence was achieved after 5 iterations. The resulting parameter set is:

$$\boldsymbol{p} = \begin{bmatrix} 0.7457 & 0.1093 & 0.0303 & 0.0187 & 0.0062 & 0.0627 \end{bmatrix}^T. \tag{5.35}$$

Evaluating the parameter set, it can be seen that the parameter turn out to be small except for the first and second one. This result complies with the assumption that especially slope dependent slip will occur in rough terrain and that rolling resistance in loose soil has a crucial impact on slip.

Validation: To validate these result, the odometry was recorded during a long range test (LRT), where a distance of $838\,\mathrm{m}$ was driven on the volcanic ground. Additionally, ground truth information was obtained using a differential global positioning system (DGPS), which was mounted on top of the rover. Figure 5.7 shows the aerial view of the test site together with the DGPS data of the driven trajectory. In the aerial view, the ROBEX base camp, a car, and the mountain road are visible and can be used for orientation. The start point of the test is close to the base camp and is denoted by (0). From there, two big clockwise circles were driven. At first, the path proceeded parallel to the road (1) and then turned downhill for approximately $100\,\mathrm{m}$ (2). On the lowest point, the rover traversed back on the volcanic terrain, once again parallel to the road (3) and then turned uphill (4) to return to the basecamp. The second circle followed the track of the first circle except at (2). Before returning to the basecamp and finishing at the start point (0), one additional, third, small circle was driven close to (1). Note that while driving the second circle, starting at (4), the DGPS lost precision and a significant error, especially in z-direction, occurred in the ground truth measurements during a distance of approximately $50\,\mathrm{m}$.

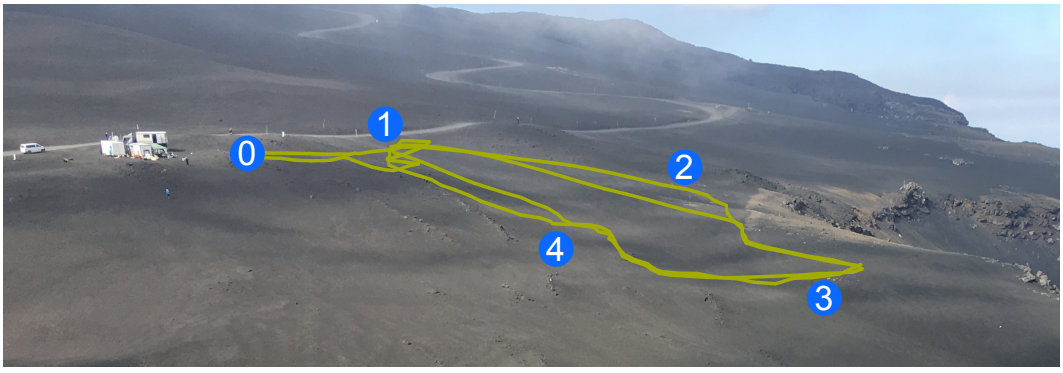

fig. 5.7.: DGPS trajectory of the long-range test mapped onto the aerial view of the flank of Mt. Etna. Additionally, the base camp and the mountain road can be seen.

Figure 5.8 shows the wheel odometry with and without slip estimation compared to the DGPS. Recall, that the current system architecture only considers the

velocities from the wheel odometry in the body-x and body-y direction. To focus on slip estimation for these relevant velocities, the plots where created by only taking these velocities from the wheel odometry and using the orientation from the EKF filter output instead.

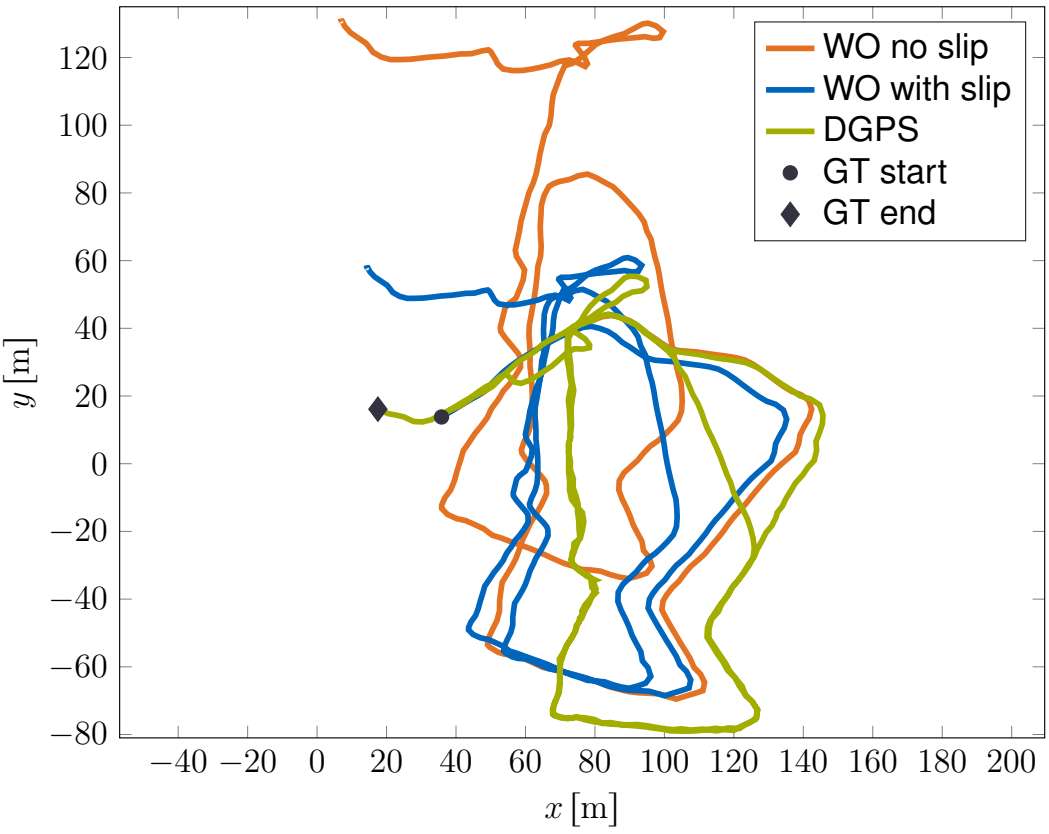

fig. 5.8.: The wheel odometry (WO) during the long-range test, compared with the DGPS data. Only the linear velocities are used from the wheel odometry, angular velocities are from the EKF output.

At the beginning of the LRT, both wheel odometries perform well but at (1) it can be seen that the slip estimation overestimates the slip due to rolling resistance causing the wheel odometry to underestimate the driven distance. However, during the remaining trajectory, especially at the uphill part (3), the wheel odometry without slip estimation significantly overestimates the driven path, whereas the new odometry provides reasonable results. The DGPS data – now displayed in the x-y-plane – clearly shows that the trajectory of the second circle mostly aligns with the first circle. While considering slip, the wheel odometry is capable to determine a trajectory where both circles almost align, which is clearly not the case for the wheel odometry neglecting slip.

Note that the circles computed by the slip-aware wheel odometry are significantly smaller than the circles of the DGPS, which is contributed to the overestimated rolling resistance with the parameter p_2. This suggests that the assumption of homogeneous soil characteristic does not hold for all areas on the mountain, as in some areas the soil is compacted stronger than in others. This

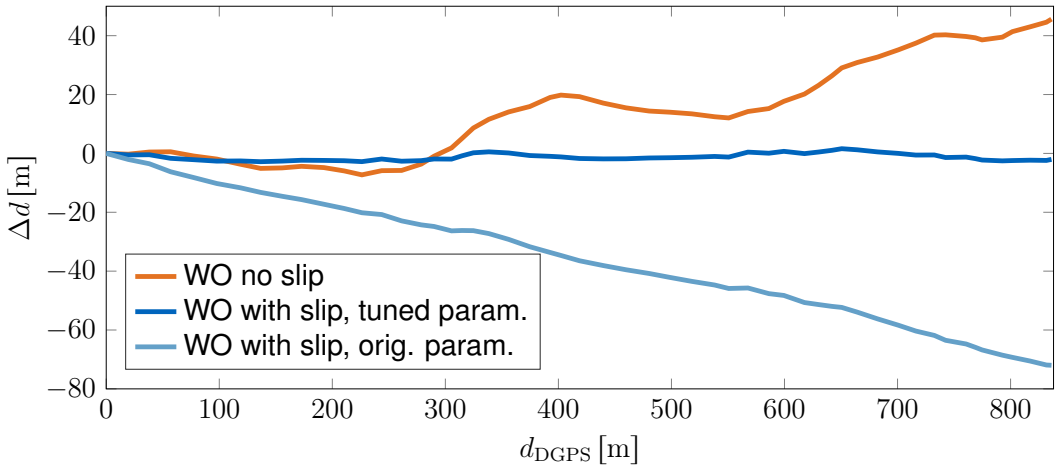

fig. 5.9.: The offset between the estimated distance of the wheel odometry (WO) and the DGPS data during the long-range test.

causes a difference in the occurring slip, which was already observed during the parameter experiments. The hypothesis for the high value of parameter p_2 is, that the parameter experiments were performed in an area with higher rolling resistance due to loose soil.

However, this observation can be used to adapt parameter p_2 by hand. Recall that the original value is 0.1093, meaning that approximately 11% of the forward velocity is lost to slip. The parameter is hand tuned to $p_2 = 0.03$ and the wheel odometry for the LRT is computed again with otherwise unchanged parameters. Figure 5.9 shows the offset between the distance measured by the DGPS and the wheel odometry. It can be seen that the wheel odometry without slip clearly overestimates the driven distance except for two downhill drives. There, the wheel odometry is not able to detect the skidding of the wheels, which causes a negative change of error in the computed distance.

As it was discussed previously, the slip-aware wheel odometry with the original parameter-set constantly overestimates the rolling resistance, causing a significant error in the computed distance. However, it is assumed that this is only contributed to the parameter p_2. Indeed, considering the distance offset with the new parameter p_2, the error is almost completely eliminated. This shows that only p_2 was determined wrongly and correcting the parameter allows to obtain an excellent computation of the driven distance with close to no error at all. The evaluation of the LRT analogue to figure 5.8 considering the adapted parameter can be found in the appendix B.1.

It can therefore be concluded that the use of slip estimation is essential for pose estimation in rough terrain. The presented results showed that the slip-aware wheel odometry allows for significant improvements in determining the velocities of the translational motion compared to a wheel odometry that does not consider slip. For future applications, it is suggested to perform the driving experiments in several different areas to account for possible changes in soil. However, if this

turns out to be non-feasible, manual fine-tuning of the parameters turned out to be an easy-to-use method with very promising results.

An additional observation during the LRT was that a possible nonlinear relation holds between the velocity and the slip velocity due to rolling resistance. It was witnessed that the wheels tend to slip slightly less at lower velocities due to velocity dependent wheel-ground interaction. However, this has little influence on the outcome of the LRT as the rover was driving at a constant speed for most of the time, but could be part of future work.

5.3.5. Simulation: Online Estimation

Another way to calibrate the set of parameters can be done by online estimation techniques. For this, [27] proposes the use of an EKF:

The state vector is the parameter vector $\boldsymbol{x}_k = \boldsymbol{p}_k$ with k denoting each parameter estimation step. Furthermore, $\boldsymbol{r}$ is the measurement residual

$$\boldsymbol{r}_k = \boldsymbol{y}_{\mathrm{meas},k} - \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-). \tag{5.36}$$

After the rover covered a certain distance, the parameters are updated in the next parameter estimation step. The EKF equations are [27]

$$\hat{\boldsymbol{x}}_k^- = \hat{\boldsymbol{x}}_{k-1}^+, \tag{5.37a}$$
$$\boldsymbol{P}_k^- = \boldsymbol{P}_{k-1}^+ + \boldsymbol{Q}_{\mathrm{step},k}, \tag{5.37b}$$
$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}_{\mathrm{sys},k}^T \left(\boldsymbol{H}_{\mathrm{sys},k} \boldsymbol{P}_k^- \boldsymbol{H}_{\mathrm{sys},k}^T + \boldsymbol{R}_{\mathrm{sys},k}\right)^{-1}, \tag{5.37c}$$
$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k \boldsymbol{r}_k \tag{5.37d}$$
$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_{\mathrm{sys},k}) \boldsymbol{P}_k^-. \tag{5.37e}$$

The covariance matrix $\boldsymbol{Q}_{\mathrm{step}}$ weights the state of the previous parameter estimation step and can be used to control the update rate. The smaller $\boldsymbol{Q}_{\mathrm{step}}$ is chosen, the smaller will be the update of the state within a parameter estimation step as the filter will rely more on the estimate of the previous step rather than the new measurement. The matrix $\boldsymbol{H}_{\mathrm{sys}}$ is the parameter Jacobian and needs to be determined following section 5.3.2 before each parameter estimation step is performed. As the parameter Jacobian is a linearization of the parameter influence on the respective part of the trajectory which is used for an update step, the resulting filter is referred to as EKF.

Additionally, the measurement covariance $\boldsymbol{R}_{\mathrm{sys}}$ needs to be found for each update step k. For this covariance, several aspects need to be considered. The ground truth measurement of the end point has a measurement covariance of $\boldsymbol{\Sigma}_{\mathrm{meas}}(t_{\mathrm{end}})$. The estimated end point depends on the accuracy of the starting point measurement $\boldsymbol{\Sigma}_{\mathrm{meas}}(t_0)$ and additionally on the sensor noise $\boldsymbol{Q}_{\mathrm{sys}}$ of the wheel odometry. These two uncertainties have influence during the whole

trajectory which is considered by applying the transition matrix. As result, the measurement covariance matrix is determined as [27]

$$\begin{aligned}\boldsymbol{R}_{\text{sys}} &= \boldsymbol{\Sigma}_{\text{meas}}(t_{\text{end}}) + \boldsymbol{\Phi}(t_{\text{end}}, t_0)\boldsymbol{\Sigma}_{\text{meas}}(t_0)\boldsymbol{\Phi}(t_{\text{end}}, t_0)^T \\ &+ \int_{t_0}^{t_{\text{end}}} \boldsymbol{\Phi}(t_{\text{end}}, \tau)\boldsymbol{G}(\tau)\boldsymbol{Q}_{\text{sys}}\boldsymbol{G}(\tau)^T\boldsymbol{\Phi}(t_{\text{end}}, \tau)^T \mathrm{d}\tau. \end{aligned} \tag{5.38}$$

The described online parameter estimation was not used during the experiments, as no online ground-truth information was directly available to the system without post-processing. Additionally, the computation of the transition matrix $\boldsymbol{\Phi}$ requires the convolution integral, which in return requires storage of the trajectory in the system memory, a task which jeopardized the real-time capabilities of the system.

However, this method was tested in simulation using the recorded data of the LRT as input. The ground truth reference was the DGPS trajectory of the LRT. Each parameter update was computed for trajectory parts with a length of approximately $4\,\mathrm{m}$. The parameters of each update step are shown in figure 5.10.

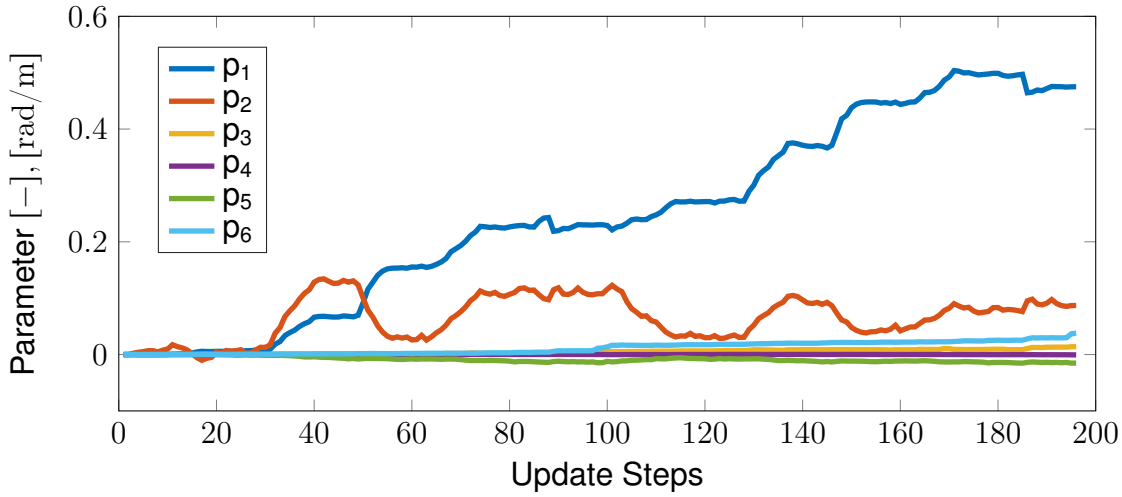

fig. 5.10.: Slip parameters as result of the online parameter estimation using DGPS as ground truth information.

It can be seen that the parameters are qualitatively similar to the parameters (5.35), especially p_{1-3} for the linear velocities. However, it can be noted, that p_2 exhibits significant jumps in its value, which supports the assumption of the non-homogeneous terrain, which has varying influence on the rolling resistance while driving.

Note that it was necessary to use very slow update rates with

$$\boldsymbol{Q}_{\text{step}} = 3 \cdot 10^{-5}\,\mathrm{diag}(1, 1, 1, 0.1, 0.1, 0.1), \tag{5.39}$$

to prevent instability of the parameter estimation. At the end of the LRT, p_1 is smaller than at the batch estimation method, but might still approach the value

of approximately 0.7 for longer tests. Generally, it can be concluded, that the EKF online parameter estimation method is very sensitive to the length of the trajectory parts, used for each update step, and the update rate.

These results show, that it is theoretically possible to use the described online calibration method on the LRU. The necessary steps would be to make ground truth information like the DGPS online available to the system, improve the computational efficiency of the code to achieve real time capability, and to study how to improve the update speed for the slip parameter.

Note that an online parameter estimation using the fusion filter output as reference would also be possible. There, a correlation between the wheel odometry and the fusion filter would still occur, but probably result to be small enough, as it is only used to tune the error-model-parameter and not to directly correct the body velocity. An investigation of the resulting correlation should nevertheless be performed.

6. Traction Control and Enhanced Slip-Estimation

Most of the traction control techniques in rough terrain focus on either optimal torque distribution or contact force maximization on the wheels. Especially slippage oriented traction control focuses on torque control, for example [30] or [16]. Recall that the mode of operation of the LRU uses the kinematic control however, making the torque as control input inaccessible.

Instead, the traction control for the LRU focuses on finding control laws for the commanded velocities and accelerations, which allow to improve the driving performance.

Recall figure 5.2 in chapter 5 comparing the stress-strain relationship of different types of soil. It is assumed that the volcanic terrain on Etna has the characteristics of a plastic soil, as it consists of small volcanic gravel with a grain size between millimeters and few centimeters. Gravel and sand with low moisture content can generally be considered as plastic soil, according to [52, p. 115]. Brittle soils on the other hand are soils like dense sands, frozen surfaces and generally compacted soils [52, p. 115].

As there is no clear maximum in the depicted shear-stress characteristic of plastic soil, no tractive optimum can be defined for traction control. Increased slip does not reduce the tractive effort of a wheel, thus having no negative impact on the driving dynamics from a tractive point of view.

However, other factors can be considered which motivate slippage reduction. For higher slip values, the tractive effort barely increases but the power P consumed rises linearly to the wheel speed with

$$P = T\omega \tag{6.1}$$

where T and ω are the wheel torque and angular wheel velocity. This illustrates that unnecessarily high slip results in unnecessary additional power consumption with very little tractive gain and motivates the application of a traction control.

Additionally, driving tests showed another negative impact of slip on the system: It turned out that for high slippage and high ω_i, the wheel interacts with the ground in such way that strong oscillations are transferred to the body. These oscillations are significantly stronger than the ones during the laboratory tests. The body-excitation negatively influences the driving performance and is a potential cause for damage to the rover.

On the LRU, only the velocities commanded to the wheels and the acceleration limiter can be used as input for traction control while the rover is operating in the kinematic control mode. Recalling chapter 5, the most obvious control attempt is to reduce the commanded acceleration on the rover. This allows to reduce the required tractive forces and therefore to reduce slip. Additionally, the commanded velocity is reduced according to the current slip, which aims at decreasing the body oscillations.

Another reason for velocity reduction is that, contrary to the model for the slip velocity (5.7), some nonlinear influences cause higher slip at higher velocities and the plastic soil assumption does not hold for all areas on Mt. Etna. It is assumed that this nonlinear influence results from the porous volcanic character of the soil, which results in a stick-slip phenomenon on a rolling wheel. Contrary to non-volcanic gravel, the grains on Mt. Etna become wedged together due to their coarse surface, which are separated only for higher relative velocities between the wheel and the ground – therefore less slip occurs at lower velocities. The stick-slip phenomenon can also be considered as the source for the body oscillations. A detailed, future investigation on the volcanic soil, its terrain properties, and the resulting driving characteristics would provide valuable information and allow for further improvements in traction control.

For the LRU operating in the kinematic control mode, the following concept for traction control is therefore presented: Use the available information on slip and reduce the driving velocity and limit the acceleration of the rover accordingly to avoid slip and strong body accelerations and to reduce energy consumption. Additionally, the control is used to compensate the understeering which was detected during the slip parameter estimation in section 5.3.

The developed traction control acts on the *desired body velocities* provided by the path planner and hands the *commanded body velocities* to the kinematic control with the inverse kinematics algorithm. The control law is governed by the slip determined in the *enhanced slip estimation*. Recall figure 2.3 for an overview on the control design.

6.1. Enhanced Slip Estimation for Traction Control

In section 5.2.2 it was discussed that information on slip could be obtained by comparing the calculated velocity provided by the EKF fusion filter with the velocity from the wheel odometry. While this method was not used for the slip estimation in chapter 5 to avoid correlations in the fusion EKF, this restriction does not hold anymore for determining slip as input for traction control.

Chapter 5 already provided information on the body slip. Instead of replacing this source of information, both types of slip estimation can be combined for an improved result. The previously mentioned body oscillations create noise in the EKF output velocity. Therefore, during the fusion process, a Kalman filter is used, which will rely primarily on the values by the slip estimation of chapter 5

and additionally use the EKF fusion as corrective influence to account for so far unobserved sources of slip.

The fusion is done by a simplified Kalman filter (see for example [48, p. 42] for a similar filter) which uses the equations of (3.49a)–(3.50c) to combine the velocity information from the EKF fusion and the velocity information from the velocity-error-model to determine the 3 DOF enhanced slip velocity $\dot{\boldsymbol{\xi}}_{\mathrm{B,slip}}$:

For the linear velocity, the measurement is obtained as

$$\boldsymbol{y} = \boldsymbol{v}_{\mathrm{B,WO}} - \boldsymbol{v}_{\mathrm{B,EKF}}, \tag{6.2}$$

and the system input is defined as

$$\boldsymbol{u} = \boldsymbol{v}_{\mathrm{B,WO,slip}}, \tag{6.3}$$

which is the slip velocity estimated by the velocity-error-model in section 5.3. The analogous equations are applied to the angular velocity.

The system matrices are chosen as $\boldsymbol{F} = 0.2\boldsymbol{I}$, $\boldsymbol{G} = 0.8\boldsymbol{I}$, and $\boldsymbol{H} = \boldsymbol{I}$. The matrices from the dynamic equation are set in such way, that the system input is strongly considered, while still relying on the values from the previous timestep for smoothing.

The covariance matrices are set to $\boldsymbol{Q} = \boldsymbol{I}$ and to $\boldsymbol{R} = 5\boldsymbol{I}$, both with the units $\mathrm{m}^2/\mathrm{s}^2$ and $\mathrm{rad}^2/\mathrm{s}^2$. The uncertainty on the EKF measurement is much higher as the signal is negatively influenced by the body oscillations.

After determining the slip velocity, the slip $|s|$ on body level is computed as

$$|s| = \begin{cases} \frac{\|\boldsymbol{v}_{\mathrm{B,slip}}\|_2}{\|\boldsymbol{v}_{\mathrm{B,WO}}\|_2} & \text{for} \quad \|\boldsymbol{v}_{\mathrm{B,WO}}\|_2 > 0.02\,\mathrm{m/s} \\ 0 & \text{otherwise} \end{cases} \tag{6.4}$$

considering the linear velocity components.

Slip is set to zero for very low velocities to avoid the singularity. Note that this is only one of many possible ways to use body slip for traction control. It would be possible to calculate a slip vector $\boldsymbol{s}$ which considers slip in all velocity components individually or alternatively to calculate slip s only in driving direction, allowing to distinguish between slipping ($s > 0$) and skidding ($s < 0$).

An example for the input signals and the filter output are shown in the Appendix. See figure B.2.

6.2. Feed-Forward Control of Velocities and Accelerations

The previous section illustrated, how slip was computed. This body slip can now be used for a feed-forward control of the LRU using as input $s_{\mathrm{control}} = |s|$.

The feed-forward control scales the desired body velocity $\boldsymbol{v}_{\mathrm{B,des}}$, which was provided by the path planner and commands the scaled velocity $\boldsymbol{v}_{\mathrm{B,cmd}}$ to the kinematic control. The kinematic control also receives an acceleration limit $a_{\max}$, which is the maximum acceleration that allowed to be applied to the body. The control algorithm for the acceleration reads

Algorithm 1 Acceleration Limiter

Input: Body velocity $\boldsymbol{v}_{\mathrm{B,withSlip}}$, slip s_{control}
if $\boldsymbol{v}_{\mathrm{B,withSlip}}$ > threshold **then**
$a_{\max} = b\, a_{\mathrm{max,motor}}(1 - c \min(s_{\mathrm{control}}, s_{\max}))$
else
$a_{\max} = a_{\mathrm{max,motor}}$
end if
Output: Acceleration limit $a_{\max}$.

Analogously, the velocity control is defined as

Algorithm 2 Scaling of the desired Velocity

Input: Body velocity $\boldsymbol{v}_{\mathrm{B,withSlip}}$, slip s_{control},
desired body velocity $\boldsymbol{v}_{\mathrm{B,des}}$ provided by the path planner.
if $\boldsymbol{v}_{\mathrm{B,withSlip}}$ > threshold **then**
$\boldsymbol{\epsilon} = [1,1,1]^T(1 - d \min(s_{\mathrm{control}}, s_{\max}))) - [0,0,1]^T p_6$
else
$\boldsymbol{\epsilon} = [1,1,1]^T - [0,0,1]^T p_6$
end if
$\boldsymbol{v}_{\mathrm{B,cmd}} = \boldsymbol{\epsilon} \circ \boldsymbol{v}_{\mathrm{B,des}}$
Output: Commanded body velocity $\boldsymbol{v}_{\mathrm{B,cmd}}$.

The control algorithms are parameterized by b, c, d, and $s_{\max}$. The wheel motors have an internal acceleration limiter $a_{\mathrm{max,motor}}$, which was determined to be $0.63\,\mathrm{m/s^2}$ and the threshold is set to be at $0.02\,\mathrm{m/s}$. The velocity control also compensates the understeering, which was detected in section 5.3 and modeled by the parameter p_6.

Note that both control laws only hold for velocities above the threshold. This design choice is due to the fact that little slip is expected at low velocities and that the calculation of slip has a singularity for stand still (see equation (5.1)). The enhanced slip estimation already accounted for te singularity. However, it is considered here again to allow for modularity between the individual odometry and control components.

6.3. Experimental Validation

To validate the derived control law, two trajectories were recorded – one with activated and one with deactivated control. The control parameters were set to $b = 0.8$, $c = 1.7$ and $d = 0.8$. To prevent very slow commanded velocities, the considered slip was limited to $s_{\max} = 0.5$. The scaling ϵ of the velocities is determined for an upper velocity limit of $5\,\mathrm{rad/s}$ for the wheels, which was set as fixed parameter for the LRU during the ROBEX demo mission. Changes in the maximum driving velocity will require to adapt the control parameters d and $s_{\max}$.

Both trajectories followed an approximately identical path, each driving a closed ellipsoid. The first part of the trajectory led straight up a slope, turned around at the highest point, and followed the slope downward past the starting point, turning again and returning to the origin. Figure 6.1 shows the result of the velocity control factor on the commanded body velocity. Between the seconds 13 to 55, the rover drives upwards and experiences strong slip. The velocity control factor is used to reduced the desired velocities according to the detected slip. The resulting commanded velocity is applied via the inverse kinematics to the wheels. Both velocities are compared in the figure 6.1.

During the remaining part of the trajectory, the rover drives downhill, where the velocity is reduced less. There is still a reduction, as the rover now slips due to skidding but less than before, as skidding and rolling resistance cancel each other out to some extend while driving on downhill slopes.

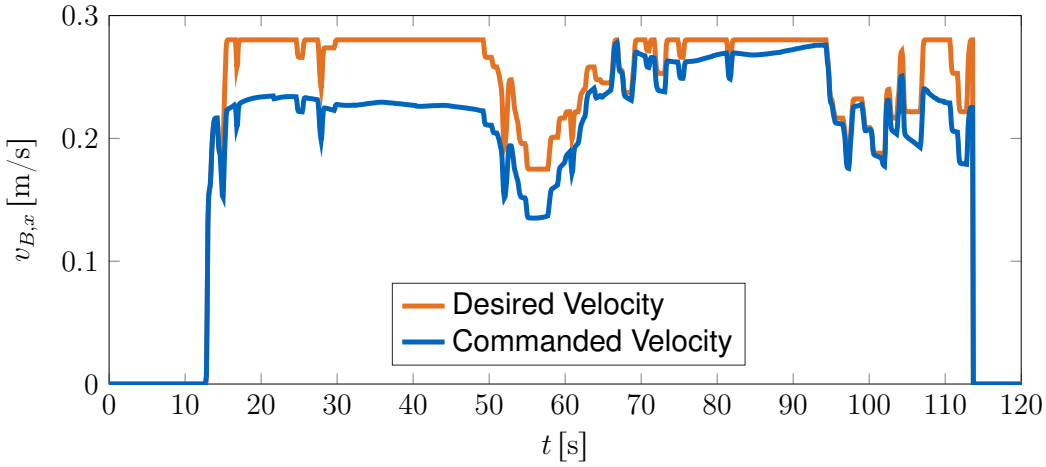

fig. 6.1.: Feed-forward control: scaling of the desired body velocities according to the estimated slip.

The velocity reduction and the acceleration limitation due to traction control has the desired impact on the slip. This can be seen in figure 6.2, which shows the slip of the trajectories with and without traction control. It has to be pointed out that the displayed slip is computed from the output of the enhanced slip velocity estimation. Unfortunately, no ground truth information was available during

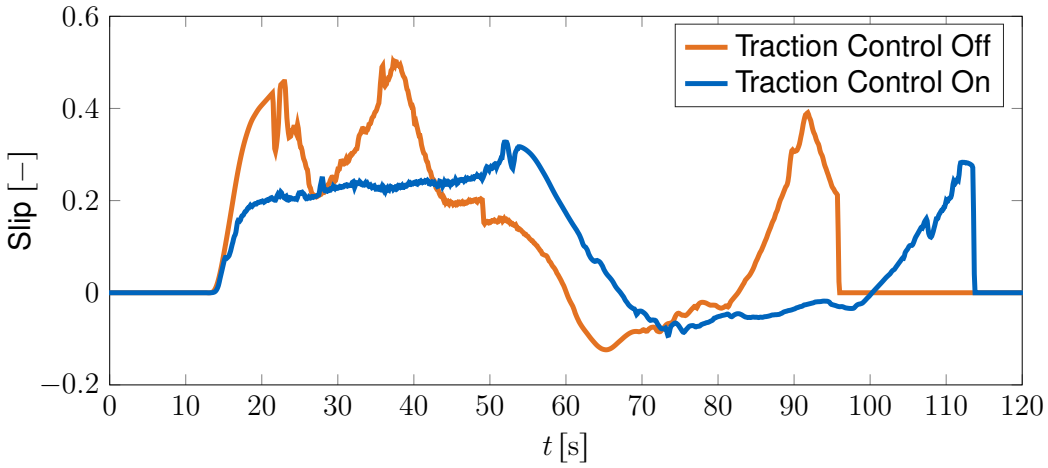

fig. 6.2.: Traction control: The effect of the control mechanism during two similar trajectories. The slip is computed by the enhanced slip estimation.

these experiments but the enhanced slip estimation is assumed to provide reasonable results. Comparing the slip shows that the traction control is capable of notable slippage reduction. This holds especially during the uphill drives, where slip is reduced significantly.

Note that with activated traction control, more time is required to drive the same trajectory. Additionally, as slip is a relative quantity, only the reduction of the slip amplitude is of relevance, not for how long slippage occurs.

Another positive effect of the traction control can be seen by evaluating the body accelerations. Figure 6.3 shows the body accelerations for trajectories without and with traction control, measured by the IMU. It can be seen that

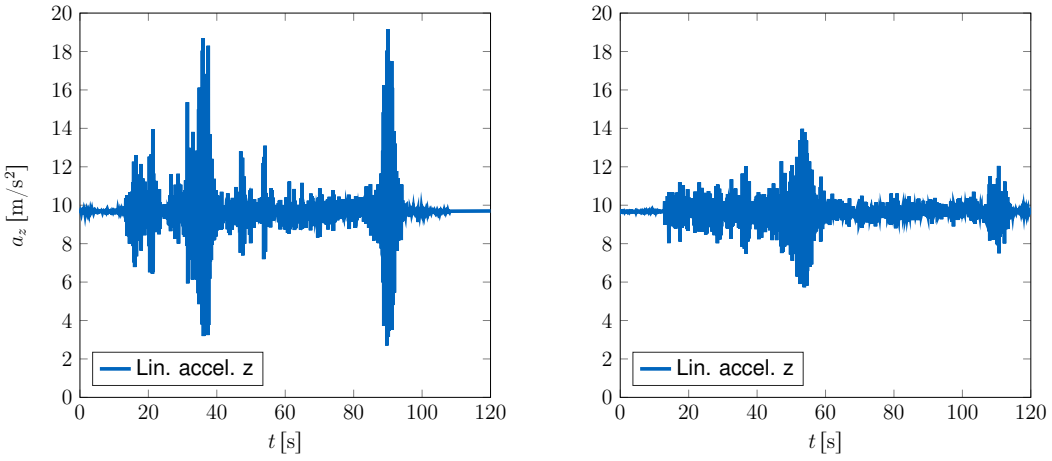

fig. 6.3.: The vertical accelerations of the rover body without (left) and with (right) active traction control, measured by the IMU.

the strongest oscillations occur at the two turning points at $35\,\mathrm{s}$ and $90\,\mathrm{s}$ or at $55\,\mathrm{s}$ and $110\,\mathrm{s}$ respectively. The first turning point is at the steepest part of the

trajectory and is one of the points, where the rover therefore experiences the most slip, which in return causes the strong body oscillations. The other point of strong oscillations is at the bottom turning point. There, the slope is less steep but the area consists of very loose soil, also causing strong slip. It can be seen that the activated traction control allows to reduce the the magnitude of body accelerations especially in moments of high slip. The peak accelerations are hereby reduced by more than 50%.

Note that the used control parameters are a first guess, with only little hand tuning afterwards. This implies, that there is even more potential for improvement by further refining the parameters, which was not possible during the ROBEX mission due to limited system time. It can be seen that the presented traction control parameters already result in substantial improvements in the driving dynamics and that the proposed control concept has some potential for further optimization.

Additionally, it has to be mentioned, that the use of slip in the control algorithms relying on the absolute value $s_{\text{control}} = |s|$ is a very conservative choice. This implies that also during downhill slopes the velocity is strongly limited to prevent skidding. Alternative approaches would be to

- consider only positive slip $s_{\text{control}} = \max(s, 0)$,
- to increase the speed while driving downhill, using $s_{\text{control}} = s$
- or to consider slip in each direction individually with a *slip vector* $\boldsymbol{s}$.

7. Conclusion

This master's thesis presented a slip-aware three-dimensional wheel odometry for a planetary exploration rover and successfully demonstrated the performance during the Mt. Etna test campaign. All development steps and design choices towards the wheel odometry are outlined and summarized in this chapter.

The thesis additionally lists several topics which are of interest for further investigation. An overview of some of the most promising topics for future research is provided at the end of this chapter.

7.1. Summary

The preceding work discussed the multitude of aspects which are relevant for a reliable wheel odometry. At first, a short description of the test scenario on the volcano Mt. Etna and an introduction of the system and its components was provided. This was followed by presenting the mathematical methods used throughout the thesis.

For the computation of the wheel odometry, the required kinematic model was derived considering all degrees of freedom of the system. A forward kinematics framework was presented which allows to obtain the velocity information of the rover body from the measured wheel velocities. The underlying assumptions and simplifications were discussed with respect to the available sensoric information.

The calibration of the rover was one important step towards increased accuracy of the wheel odometry. Many different calibration steps were applied, including the measurement of the real rover geometry, the modeling of the wheel elasticity as rotational spring, or selection of sensoric inputs that are less exposed to measurement noise.

The wheel odometry is influenced by an additional major error source once the rover enters rough terrain: the wheel slip. At first, methods for direct measurement of the slip were investigated. It turned out though, that the system setup of the LRU favors slip estimation rather than direct slip measurements.

This thesis adapted an existing method for slip estimation to the LRU by introducing a parameterized velocity-error-model for body slip. The terrain-dependent parameters of the model were determined by driving experiments in rough terrain on Mt. Etna. The slip model and its parameters were validated during a

long-range driving test on the volcano, covering almost one kilometer in distance.

Finally, the obtained information on slip was used for traction control. The special feature of the proposed traction controller is that it acts on the body velocities and accelerations instead of adapting the wheel torques. It was demonstrated experimentally, that this approach is not only capable to reduce slippage but also increases the overall driving performance.

7.2. Outlook

During the course of this work, several areas for further improvement were discussed. First of all, to achieve full three-dimensional wheel odometry, new attempts on wheel contact angle estimation can be applied. A promising concept would be to use the visual odometry as source for information on obstacles and the slope and use this to determine the wheel contact angles.

Due to operational constraints with regard to the ROBEX Moon Analogue Mission, only static covariances for the wheel odometry have been used for sensor fusion during the experiments on Mt. Etna. Changing the covariances from the so far hand-tuned values to the proposed dynamic covariances is potentially beneficial to the pose estimation. However, this will strongly influence the pose estimation filter and therefore requires thorough testing.

The method of slip parameter estimation can be upgraded to the improved two-dimensional case, where the instantaneous plane of movement of the rover is fitted locally to the slope, once there are means available for three-dimensional ground truth measurements. Then, even the 6 DOF slip parameter estimation model could be used, assuming that there is enough computational power to determine the parameter Jacobian matrices numerically.

Additionally, the online slip parameter estimation can be further investigated. Solving the stability issues and investigating the correlation with the EKF, an online parameter estimation using the fusion EKF output as ground truth reference might result in a powerful and easy-to-use method for slip estimation.

The long-range driving experiment suggested that the applied method does provide a reasonable, but not yet optimal, set of slip parameters. Possible changes in the experimental setup could be investigated. Additionally, more complex slip-velocity-error-models could be applied. Especially a nonlinear modeling of the rolling resistance could be considered, as this relation is suggested by observations during the long-range test and the traction control experiments.

A. Bibliography

[1] Bilstein R., *Orders of Magnitude: A History of the NACA and NASA, 1915-1990*. National Aeronautics and Space Administration, 1989.

[2] National Aeronautics and Space Administration (NASA), "Mars missions." Webpage, accessed 18 July 2017. https://solarsystem.nasa.gov/missions/target/mars.

[3] Bajracharya M., Maimone M., and Helmick D., "Autonomy for mars rovers: Past, present, and future," *Computer*, vol. 41, pp. 44–50, 2008.

[4] Wedler A., Rebele B., Reill J., Suppa M., Hirschmüller H., Brand C., Schuster M., Vodermayer B., Gmeiner H., Maier A., Willberg B., Bussmann K., Wappler F., and Hellerer M., "LRU – Lightweight Rover Unit," in *13th Symposium on Advanced Space Technologies in Robotics and Automation: ASTRA*, 2015.

[5] Alfred-Wegener-Institute | Helmholtz-Center for Polar- and Marine Research, "Robex – Robotic Exploration of Extreme Environments." Webpage, accessed 15 July 2017. http://www.robex-allianz.de/.

[6] Schuster M., Brand C., Brunner S., Lehner P., Reill J., Riedel S., Bodenmüller T., Bussmann K., Büttner S., Dömel A., Friedl W., Grixa I., Hellerer M., Hirschmüller H., Kassecker M., Marton Z., Nissler C., Ruess F., Suppa M., and Wedler A., "The LRU rover for autonomous planetary exploration and its success in the spacebotcamp challenge," in *International Conference on Autonomous Robot Systems and Competitions: ICARSC*, pp. 7–14, 2016.

[7] Gong W., "Discussions on localization capabilities of MSL and MER rovers," *Annals of GIS*, vol. 21, pp. 69–79, 2015.

[8] Siegwart R. and Nourbakhsh I., *Introduction to Autonomous Mobile Robots*. The MIT Press, 2004.

[9] Siciliano B. and Khatib O., *Springer Handbook of Robotics*. Springer, 2008.

[10] Blair S., "Rovers return," *Engineering and Technology*, vol. 6, pp. 48–50, 2011.

[11] Xiao L., “China’s touch on the moon,” *Nature Geoscience*, vol. 7, pp. 391–392, 2014.

[12] European Space Agency (ESA), “Exomars rover.” Webpage, accessed 17 July 2017. http://exploration.esa.int/mars/45084-exomars-rover.

[13] Tarokh M., McDermott G., Hayati S., and Hung J., “Kinematic modeling of a high mobility mars rover,” in *IEEE Int. Conf. Robot. Autom.: ICRA*, vol. 2, pp. 992–998, 1999.

[14] Helmick D., Cheng Y., Clouse D., Bajracharya M., Matthies L., and Roumeliotis S., “Slip compensation for a mars rover,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems: IROS*, pp. 2806–2813, 2005.

[15] Seegmiller N., Rogers-Marcovitz F., and Kelly A., “Online calibration of vehicle powertrain and pose estimation parameters using integrated dynamics,” in *IEEE Int. Conf. Robot. Autom.: ICRA*, pp. 3969–3974, 2012.

[16] Lamon P., Krebs A., Lauria M., Siegwart R., and Shooter S., “Wheel torque control for a rough terrain rover,” in *IEEE Int. Conf. Robot. Autom.: ICRA*, pp. 4682–4687, 2004.

[17] Sakai A., Tamura Y., and Kuroda Y., “An efficient solution to 6DOF localization using unscented kalman filter for planetary rovers,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems: IROS*, pp. 4154–4159, 2009.

[18] Ojeda L., Cruz D., Reina G., and Borenstein J., “Current-based slippage detection and odometry correction for mobile robots and planetary rovers,” *IEEE Trans. Robot.*, vol. 22, pp. 366–378, 2006.

[19] Muir P. and Neuman C., “Kinematic modeling of wheeled mobile robots,” *J. Robotic Syst.*, vol. 4, pp. 281–340, 1987.

[20] Tarokh M. and McDermott G., “A systematic approach to kinematics modeling of high mobility wheeled rovers,” in *IEEE Int. Conf. Robot. Autom.: ICRA*, pp. 4905–4910, 2007.

[21] Seegmiller N. and Kelly A., “Enhanced 3D kinematic modeling of wheeled mobile robots,” in *Robotics: Science and Systems X*, 2014.

[22] Baumgartner E., Aghazarian H., Trebi-Ollennu A., Huntsberger T., and Garrett M., “State estimation and vehicle localization for the FIDO rover,” in *Proceedings of SPIE*, vol. 4196, pp. 329–336, 2000.

[23] Lamon P. and Siegwart R., “Inertial and 3D-odometry fusion in rough terrain - towards real 3D navigation,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems: IROS*, Institute of Electrical and Electronics Engineers (IEEE), 2004.

[24] Borenstein J., Everett H., and Feng L., *Where am I? Sensors and methods for mobile robot positioning*. Univ. of Michigan, 1996.

[25] Ojeda L. and Borenstein J., "Methods for the reduction of odometry errors in over-constrained mobile robots," *Autonomous Robots*, vol. 16, pp. 273–286, 2004.

[26] Kelly A., "Linearized error propagation in odometry," *Int. J. Robot. Res.*, vol. 23, pp. 179–218, 2004.

[27] Seegmiller N., Rogers-Marcovitz F., Miller G., and Kelly A., "Vehicle model identification by integrated prediction error minimization," *Int. J. Robot. Res.*, vol. 32, pp. 912–931, 2013.

[28] Kim J. and Lee J., "Real-time prediction of optimal control parameters for mobile robots based on estimated strength of ground surface," *Journal of Institute of Control, Robotics and Systems*, vol. 20, pp. 58–69, 2014.

[29] Kim J. and Lee J., "A kinematic-based rough terrain control for traction and energy saving of an exploration rover," in *IEEE/RSJ International Conference on Intelligent Robots and Systems: IROS*, 2016.

[30] Yoshida K., Hamano H., and Watanabe T., *Slip-based Traction Control of a Planetary Rover*, pp. 644–653. Springer, 2003.

[31] Lindgren D., Hague T., Smith P., and Marchant J., "Relating torque and slip in an odometric model for an autonomous agricultural vehicle," *Autonomous Robots*, vol. 13, pp. 73–86, 2002.

[32] Iagnemma K. and Dubowsky S., "Traction control of wheeled robotic vehicles in rough terrain with application to planetary rovers," *Int. J. Robot. Res.*, vol. 23, pp. 1029–1040, 2004.

[33] Wedler A., Hellerer M., Rebele B., Gmeiner H., Vodermayer B., Bellmann T., Barthelmes S., Rosta R., Lange C., Witte L., Schmitz N., Knapmeyer M., Czeluschke A., Thomsen L., Waldmann C., Flögel S., Wilde M., and Takei Y., "ROBEX – components and methods for the planetary exploration demonstration mission," in *13th Symposium on Advanced Space Technologies in Robotics and Automation: ASTRA*, 2015.

[34] Alfred-Wegener-Institute | Helmholtz-Center for Polar- and Marine Research, "Robex – demo missions." Webpage, accessed 15 July 2017. http://www.robex-allianz.de/en/about-robex/demo-missions.

[35] Schmid K., Ruess F., and Burschka D., "Local reference filter for life-long vision aided inertial navigation," in *International Conference on Information Fusion*, 2014.

[36] Kalman R. and Bucy R., "New results in linear filtering and prediction theory," *Journal of Basic Engineering*, vol. 83, pp. 95–108, 1961.

[37] Brammer K. and Siffling G., *Stochastische Grundlagen des Kalman-Bucy-Filters*. R. Oldenbourg Verlag München Wien, 1990.

[38] Grewal M. and Andrews A., *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley-IEEE Press, 2008.

[39] Kalman R., "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[40] Simon D., *Optimal State Estimation*. Wiley-Interscience, 2006.

[41] Ben-Israel A. and Greville T., *Generalized Inverses*. Springer, 2003.

[42] Doty K., Melchiorri C., and Bonivento C., "A theory of generalized inverses applied to robotics," *Int. J. Robot. Res.*, vol. 12, pp. 1–19, 1993.

[43] Eldén L., "A weighted pseudoinverse, generalized singular values, and constrained least squares problems," *BIT Numerical Mathematics*, vol. 22, pp. 487–502, 1982.

[44] Welch G. and Bishop G., *An Introduction to the Kalman Filter*. Univ. of North Carolina, 2006. Technical Report.

[45] Thrun S., Burgard W., and Fox D., *Probabilistic Robotics*. The MIT Press, 2005.

[46] Murray R., Li Z., and Sastry S., *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[47] Craig J., *Introduction to Robotics: Mechanics and Control*. Pearson Prentice Hall, 2004.

[48] Iagnemma K. and Dubowsky S., *Mobile Robots in Rough Terrain*. Springer, 2004.

[49] Borenstein J. and Feng L., "Correction of systematic odometry errors in mobile robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems: IROS*, pp. 569–574, 1995.

[50] German Aerospace Center (DLR), "Datasheet LRU Rover." DLR internal document, 2016.

[51] Wong J. Y., *Theory of Ground Vehicles*. Wiley, 1978.

[52] Bekker M., *Introduction to Terrain-Vehicle Systems*. Univ. of Michigan Press, 1969.

[53] Featherstone R., *Robot Dynamics Algorithms*. Springer, 2014.

B. Appendix

B.1. Slip Estimation

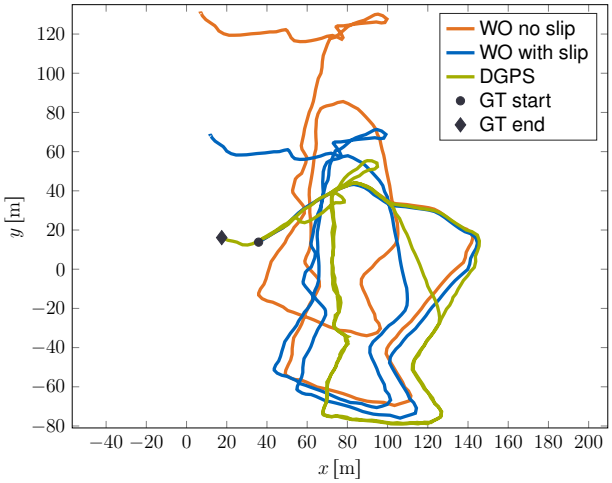

fig. B.1.: The wheel odometry (WO) during the long-range test, compared with the DGPS data. Only the linear velocities are used from the wheel odometry, the orientation is from the EKF output. The parameter p_2 is hand-tuned and set to 0.03. Note that from this perspective, the position error is mostly attributed to an error in the pose of the EKF.

B.2. Enhanced Slip Estimation

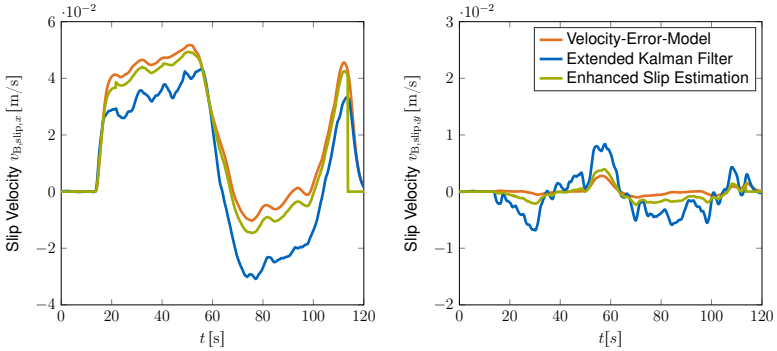

fig. B.2.: Comparison of the slip velocitiy from different sources. Note that the slip parameter $p_2 = 0.1092$ was used. The Velocity-Error-Model provides otherwise a precise slip estimate in x. The slip in y-direction is underestimated by the model compared to the EKF output.